

EDISON KLAFKE FILLUS

**CAAMPI: UMA ABORDAGEM BASEADA EM TÉCNICAS  
DE AGRUPAMENTO PARA MINERAÇÃO DE ASPECTOS  
E IDENTIFICAÇÃO DE PONTOS DE CORTE**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Profa. Dra. Silvia Regina Vergilio

CURITIBA

2012

EDISON KLAFKE FILLUS

**CAAMPI: UMA ABORDAGEM BASEADA EM TÉCNICAS  
DE AGRUPAMENTO PARA MINERAÇÃO DE ASPECTOS  
E IDENTIFICAÇÃO DE PONTOS DE CORTE**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Profa. Dra. Silvia Regina Vergilio

CURITIBA

2012

## SUMÁRIO

<b>RESUMO</b>	<b>iv</b>
<b>ABSTRACT</b>	<b>v</b>
<b>LISTA DE FIGURAS</b>	<b>vi</b>
<b>LISTA DE TABELAS</b>	<b>ix</b>
<b>1 INTRODUÇÃO</b>	<b>1</b>
1.1 Contexto . . . . .	1
1.2 Motivação . . . . .	2
1.3 Objetivos . . . . .	3
1.4 Organização . . . . .	4
<b>2 ORIENTAÇÃO A ASPECTOS</b>	<b>5</b>
2.1 Limitações da orientação a objetos . . . . .	5
2.2 Reduzindo limitações com a Orientação a Aspectos . . . . .	9
2.3 Linguagens orientadas a aspectos . . . . .	10
2.4 Introdução a Linguagem AspectJ . . . . .	10
2.5 Considerações Finais . . . . .	13
<b>3 ANÁLISE DE AGRUPAMENTO</b>	<b>14</b>
3.1 Introdução a Análise de Agrupamento . . . . .	14
3.1.1 Processo de Análise de Agrupamento . . . . .	14
3.2 Medidas de Distância . . . . .	16
3.3 Tipos de Algoritmos de Agrupamento . . . . .	16
3.4 Considerações Finais . . . . .	21
<b>4 MINERAÇÃO DE ASPECTOS</b>	<b>22</b>
4.1 Introduzindo aspectos em sistemas . . . . .	22
4.2 Identificando aspectos . . . . .	23
4.3 Minerando aspectos . . . . .	24
4.3.1 Análise de <i>Fan-in</i> . . . . .	25
4.3.2 Detecção de Clones . . . . .	25
4.3.3 Análise de repositórios de sistemas de controle de versões . . . . .	26
4.3.4 Identificação baseada em catálogo de padrões . . . . .	27
4.3.5 Classificação de métodos através de aprendizado de máquina . . . . .	27
4.3.6 Análise de padrões recorrentes em rastros de execução . . . . .	28
4.3.7 Análise de conceito formal . . . . .	28
4.3.8 Processamento de linguagem natural no código fonte . . . . .	29
4.3.9 Análise de Ligações . . . . .	30
4.3.10 Detecção de métodos únicos . . . . .	30
4.3.11 Baseada em grafos . . . . .	31
4.3.12 Análise de Agrupamento . . . . .	31
4.4 Aplicando análise de agrupamento na mineração de aspectos . . . . .	31

4.4.1	Definição formal . . . . .	32
4.4.2	Processo . . . . .	32
4.4.3	Abordagens existentes . . . . .	33
4.5	Avaliando a qualidade da mineração de aspectos . . . . .	40
4.6	Resultados encontrados na literatura . . . . .	43
4.7	Limitações identificadas . . . . .	43
4.8	Considerações Finais . . . . .	44
<b>5</b>	<b>CAAMPI: UMA ABORDAGEM BASEADA EM AGRUPAMENTO PARA MINERAÇÃO DE ASPECTOS E IDENTIFICAÇÃO DE PON- TOS DE CORTE</b>	<b>46</b>
5.1	Processo CAAMPI . . . . .	46
5.2	Medida de Distância SOND . . . . .	49
5.2.1	Medida de Distância de Espalhamento . . . . .	50
5.2.2	Medida de Distância de Operações . . . . .	50
5.2.3	Medida de Distância de Nomes . . . . .	51
5.3	Medidas de Ordenação de Grupos . . . . .	53
5.3.1	Medida de Ordenação de Grupos GSRank . . . . .	54
5.3.2	Medida de Ordenação de Grupos GSIRank . . . . .	54
5.3.3	Medida de Ordenação de Grupos GFRank . . . . .	55
5.3.4	Medida de Ordenação de Grupos GFIRank . . . . .	55
5.4	ARPI: Um método automatizado de identificação de pontos de corte . . . . .	56
5.5	CAAMPI4J: Um <i>framework</i> de implementação da CAAMPI para Java . . . . .	60
5.6	Trabalhos Relacionados . . . . .	71
5.7	Considerações Finais . . . . .	72
<b>6</b>	<b>AVALIAÇÃO DA ABORDAGEM CAAMPI</b>	<b>74</b>
6.1	Metodologia . . . . .	74
6.1.1	Objetivo dos Experimentos . . . . .	74
6.1.2	Sistemas utilizados . . . . .	74
6.1.3	Índices de avaliação de qualidade de agrupamento . . . . .	75
6.1.4	Índices de avaliação de qualidade de ordenação de grupos . . . . .	76
6.1.5	Índices de avaliação de qualidade de identificação de pontos de corte . . . . .	79
6.1.6	Etapas dos experimentos . . . . .	81
6.2	Resultados e Análise da Fase de Agrupamento . . . . .	85
6.3	Resultados e Análise da Fase de Ordenação de Grupos . . . . .	89
6.4	Resultados e Análise da Fase de Identificação de Pontos de Corte . . . . .	91
6.5	Ameaças à Validade . . . . .	92
6.6	Considerações Finais . . . . .	93
<b>7</b>	<b>CONCLUSÃO</b>	<b>94</b>
7.1	Trabalhos Futuros . . . . .	95
	<b>BIBLIOGRAFIA</b>	<b>103</b>
	<b>A INSTÂNCIAS DE INTERESSES TRANSVERSAIS VALIDADAS</b>	<b>104</b>
	<b>B CALIBRAÇÃO DAS INSTÂNCIAS DE AGRUPAMENTO</b>	<b>109</b>

C AGRUPAMENTO E ORDENAÇÃO	131
D PONTOS DE CORTE IDENTIFICADOS	166

## RESUMO

A programação Orientação a Aspectos (AO) visa a solucionar alguns problemas da orientação a objetos (OO) relativos aos chamados interesses transversais, tais como funcionalidades duplicadas, espalhamento de funcionalidades e funcionalidades entrelaçadas. Os aspectos modularizam requisitos não funcionais que precisam estar presentes em várias partes do sistema e são encaixados conforme a necessidade por pontos de corte nos objetos a serem afetados. Para usufruir destes benefícios é necessário um processo de identificação e separação dos interesses base e transversais, o que é uma tarefa árdua e trabalhosa. A Mineração de Aspectos tem como objetivo automatizar este processo. Várias técnicas para identificar automaticamente os interesses transversais existem. Dentre estas, destaca-se a análise de agrupamento que é capaz de descobrir possíveis candidatos a aspectos sem que seja necessário o conhecimento prévio das particularidades do sistema. Entretanto, a maioria das abordagens baseadas nesta técnica não utilizam uma medida de distância que considera diferentes características associadas aos sintomas de interesses transversais. Além disso, as abordagens carecem de instrumentos que permitam a eliminação de interesses base ou a rápida separação dos potenciais interesses transversais. Outra limitação é que elas não fornecem auxílio na identificação de pontos de corte. Considerando este fato, o objetivo principal deste trabalho é contribuir para a área de mineração de aspectos em código fonte baseada em análise de agrupamento através da proposição da abordagem CAAMPI, que contempla a definição de uma medida de distância que combina os sintomas de espalhamento, código duplicado e convenção de nomes. Além disso, a abordagem utiliza um filtro baseado na técnica de *fan-in* para a eliminação parcial de interesses base e permite a identificação de pontos de corte utilizando regras de associação e a ordenação dos grupos obtidos na fase de agrupamento de acordo com seu potencial de pertencer a um interesse transversal. O trabalho também descreve o *framework* CAAMPI4J, que define e implementa a abordagem CAAMPI para a análise de código fonte Java. O CAAMPI4J foi utilizado em experimentos com três sistemas reais, nos quais foram avaliadas diferentes combinações de medidas e tipos de algoritmos de agrupamento. Os resultados dos experimentos demonstram que a medida de distância proposta obteve melhores resultados que as medidas existentes, e que os pontos de corte identificados são realmente aproveitados nas versões refatoradas para aspectos dos sistemas OO utilizados. Além disso, a ordenação de grupos viabiliza a definição de critérios quantitativos para a priorização da análise e refatoração dos candidatos a aspectos.

## ABSTRACT

The aspect-oriented (AO) programming aims at solving some of the object-oriented (OO) problems related to the called crosscutting concerns, such as code cloning, scattering and tangling. The aspects modularizes non-functional requirements that need to be spread in several parts of the system, and can be joined by pointcuts on objects that are affected by these concerns. To benefit from AO, the software needs to go through a concern decomposition process, in which the core and crosscutting concerns are identified and separated. This is a hard and labour-intensive task. The goal of the Aspect Mining field is to automate this process. To do this, different techniques exist. Among them the technique based on clustering analysis is able to discover potential aspects associated to the existing crosscutting concerns without requiring prior knowledge of the software characteristics. However, the approaches based on this technique do not use a distance measure that considers different characteristics of the crosscutting concern symptoms. Moreover, they do not allow neither the elimination of base concerns nor the isolation of potential crosscutting concerns. Other limitation is that they do not help in the identification of the point cuts. Due to this fact, the main objective of this work is to contribute for the aspect mining area based on clustering analysis by proposing the CAAMPI approach, which introduces a distance measure that combines scattering, code cloning and naming convention symptoms. Moreover, the approach adopts fan-in analysis for partially eliminating core concerns. It allows the identification of pointcuts by using association rules, and the ranking of groups obtained in the cluster analysis that potentially belong to a crosscutting concern. The work also describes the framework CAAMPI4J, for Java source code analysis. The CAAMPI4J was used in experiments with three real systems, which evaluated different combinations of distance measures and clustering algorithms. The experimental results show that the proposed distance measure obtains better results than the most used distance measures, and the identified pointcuts are in fact implemented in the AO versions of the systems. In addition to this, the group ranking enables the definition of quantitative criteria for aspect candidates prioritization and refactoring.

## LISTA DE FIGURAS

2.1	Interesses transversais cruzam várias camadas de um software . . . . .	6
2.2	Espalhamento de código devido a código duplicado . . . . .	6
2.3	Espalhamento de código devido a código complementar . . . . .	7
2.4	Funcionalidade de log espalhada por vários módulos no Apache Tomcat 4 .	7
2.5	Módulo com sintoma de código entrelaçado (adaptada de [39]). . . . .	8
3.1	Processo de Análise de Agrupamento (adaptada de [69]). . . . .	15
3.2	Agrupamento em estrutura hierárquica (adaptada de [69]). . . . .	19
3.3	Processo do algoritmo CHAMELEON (adaptada de [33]). . . . .	20
4.1	Processo de decomposição e integração de interesses de um software . . . .	23
4.2	Migração de sistemas legados para aspectos . . . . .	24
5.1	Processo da Abordagem CAAMPI . . . . .	47
5.2	Diagrama de Classes da Fase de Computação do CAAMPI4J . . . . .	61
5.3	Exemplo de extração das informações de um método do JHotDraw . . . . .	61
6.1	Comparação do resultado das medidas de ordenação . . . . .	90



## LISTA DE TABELAS

4.1	Comparação do processo de Mineração de Aspectos e Agrupamento . . . . .	32
4.2	Comparação dos algoritmos existentes na literatura . . . . .	43
5.1	Resumo das contribuições da CAAMPI . . . . .	47
5.2	Exemplo de filtro de regras entre métodos adjuntos . . . . .	57
5.3	Exemplo de classificação de regras . . . . .	58
5.4	Regras Obtidas no Exemplo . . . . .	65
5.5	Classificação das relações encontradas . . . . .	66
5.6	Candidatos a Pontos de Corte Identificados . . . . .	66
5.7	Candidatos a Pontos de Corte Agrupados por Método . . . . .	66
5.8	Vinculação dos Pontos de Corte aos Grupos . . . . .	67
5.9	Exemplo Resultado CAAMPI4J . . . . .	67
6.1	Instâncias avaliadas . . . . .	81
6.2	Parâmetros Calibrados . . . . .	82
6.3	Melhor configuração de parâmetros das instâncias . . . . .	83
6.4	Resultado das instâncias por sistema . . . . .	85
6.5	Melhores medidas de distância e algoritmos de agrupamento por sistema . . . . .	86
6.6	Melhor algoritmo de agrupamento por medida de distância . . . . .	86
6.7	Melhor medida de distância por algoritmo de agrupamento . . . . .	87
6.8	Resultados das Medidas de Ordenação de Grupos . . . . .	89
6.9	Comparação do resultado das medidas de ordenação . . . . .	89
6.10	Visão dos resultados com ênfase nos algoritmos . . . . .	90
6.11	Resultado da Identificação de Pontos de Corte com ARPIM . . . . .	91
A.1	Instâncias de interesses transversais validadas no JHotDraw . . . . .	104
A.2	Mapeamento das instâncias validadas do JHotDraw com aspectos refatorados no AJHotDraw . . . . .	105
A.3	Instâncias de interesses transversais validadas no Tomcat . . . . .	106
A.4	Instâncias de interesses transversais validadas no HSQLDB . . . . .	108
A.5	Mapeamento das instâncias validadas do HSQLDB com aspectos refatorados no AJHSQLDB . . . . .	108
B.1	Calibração da Instância 1: JHotDraw . . . . .	109
B.2	Calibração da Instância 1: Tomcat . . . . .	109
B.3	Calibração da Instância 1: HSQLDB . . . . .	109
B.4	Calibração da Instância 2: JHotDraw . . . . .	109
B.5	Calibração da Instância 2: Tomcat . . . . .	109
B.6	Calibração da Instância 2: HSQLDB . . . . .	109
B.7	Calibração da Instância 3: JHotDraw . . . . .	109
B.8	Calibração da Instância 3: Tomcat . . . . .	109
B.9	Calibração da Instância 3: HSQLDB . . . . .	109
B.10	Calibração da Instância 4: JHotDraw . . . . .	110
B.11	Calibração da Instância 4: Tomcat . . . . .	110
B.12	Calibração da Instância 4: HSQLDB . . . . .	110
B.13	Calibração da Instância 5: JHotDraw . . . . .	110

B.14	Calibração da Instância 5: Tomcat	110
B.15	Calibração da Instância 5: HSQLDB	110
B.16	Calibração da Instância 6: JHotDraw	111
B.17	Calibração da Instância 6: Tomcat	112
B.18	Calibração da Instância 6: HSQLDB	113
B.19	Calibração da Instância 7: JHotDraw	114
B.20	Calibração da Instância 7: Tomcat	114
B.21	Calibração da Instância 7: HSQLDB	114
B.22	Calibração da Instância 8: JHotDraw	114
B.23	Calibração da Instância 8: Tomcat	114
B.24	Calibração da Instância 8: HSQLDB	114
B.25	Calibração da Instância 9: JHotDraw	115
B.26	Calibração da Instância 9: Tomcat	115
B.27	Calibração da Instância 9: HSQLDB	115
B.28	Calibração da Instância 10: JHotDraw	115
B.29	Calibração da Instância 10: Tomcat	115
B.30	Calibração da Instância 10: HSQLDB	115
B.31	Calibração da Instância 11: JHotDraw	115
B.32	Calibração da Instância 11: Tomcat	115
B.33	Calibração da Instância 11: HSQLDB	116
B.34	Calibração da Instância 12: JHotDraw	116
B.35	Calibração da Instância 12: Tomcat	117
B.36	Calibração da Instância 12: HSQLDB	118
B.37	Calibração da Instância 13: JHotDraw	119
B.38	Calibração da Instância 13: Tomcat	119
B.39	Calibração da Instância 13: HSQLDB	120
B.40	Calibração da Instância 14: JHotDraw	120
B.41	Calibração da Instância 14: Tomcat	120
B.42	Calibração da Instância 14: HSQLDB	120
B.43	Calibração da Instância 15: JHotDraw	120
B.44	Calibração da Instância 15: Tomcat	120
B.45	Calibração da Instância 15: HSQLDB	121
B.46	Calibração da Instância 16: JHotDraw	121
B.47	Calibração da Instância 16: Tomcat	121
B.48	Calibração da Instância 16: HSQLDB	121
B.49	Calibração da Instância 17: JHotDraw	121
B.50	Calibração da Instância 17: Tomcat	122
B.51	Calibração da Instância 17: HSQLDB	122
B.52	Calibração da Instância 18: JHotDraw	123
B.53	Calibração da Instância 18: Tomcat	125
B.54	Calibração da Instância 18: HSQLDB	128
C.1	Lista de métodos do JHotDraw	131
C.2	Lista de grupos da instância 12 com pontuação: JHotDraw	138
C.3	Lista de métodos do Tomcat	142
C.4	Lista de grupos da instância 12 com pontuação: Tomcat	148
C.5	Lista de métodos do HSQLDB	155
C.6	Lista de grupos da instância 12 com pontuação: HSQLDB	159

D.1	Pontos de Corte identificados pelo ARPIM no sistema JHotDraw . . . . .	166
D.2	Pontos de Corte identificados pelo ARPIM no sistema Tomcat . . . . .	167
D.3	Pontos de Corte identificados pelo ARPIM no sistema HSQLDB . . . . .	167

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 Contexto

A Orientação a Objetos (OO) modela os problemas do mundo real através de objetos que encapsulam seu próprio estado e comportamento. Aliada aos padrões de projetos proporciona que os sistemas sejam divididos em camadas com funcionalidades e comportamentos próprios. Este modelo consegue representar os requisitos funcionais de maneira muito eficaz, porém, este não prevê a representação dos requisitos não funcionais, que devido a sua natureza, estão presentes em várias camadas do sistema, gerando os chamados interesses transversais. A inclusão de interesses transversais acarreta em problemas de modelagem como funcionalidades duplicadas, espalhamento de funcionalidades e funcionalidades entrelaçadas, o que prejudica o entendimento do sistema e dificulta a sua manutenção e evolução.

De forma a proporcionar uma maneira de encapsular os requisitos não funcionais, foi criada a Orientação a Aspectos (OA) [37], que tem por premissa modularizar os interesses transversais em componentes chamados aspectos. Os aspectos encapsulam os dados e o comportamento esperado do interesse transversal e podem ser encaixados conforme a necessidade aos objetos que serão afetados por este interesse.

Os aspectos trazem vários benefícios ao desenvolvimento de software, como módulos com implementação mais clara e objetiva, maior modularização, facilidade de evolução e manutenção, além do aumento do reúso de código e a redução de custos e prazos de entrega [39]. Isto torna a orientação a aspectos uma opção atraente para a melhoria de todo o processo de desenvolvimento quando comparada a utilização única e exclusiva da orientação a objetos.

Para que um software possa usufruir dos benefícios da orientação a aspectos, é necessário que ele passe por um processo de decomposição de funcionalidades de forma a identificar quais funcionalidades fazem parte de interesses base e quais fazem parte de interesses transversais. Os interesses transversais identificados são então implementados em módulos individuais como aspectos, que ao final são combinados com os interesses base através da definição de pontos de corte, que indicam o ponto exato onde deve ocorrer a junção com os interesses base.

Todavia, o processo de decomposição dos sistemas para a identificação dos interesses transversais é uma tarefa árdua e trabalhosa. Visando a reduzir o esforço necessário e automatizar algumas das tarefas envolvidas neste processo, surgiram três áreas de pesquisa [34]: (i) Mineração de Aspectos Antecipada, que identifica candidatos a interesses transversais a partir da mineração de documentos de especificação do sistema; (ii) Navegadores Dedicados, que permitem a busca de candidatos a interesses transversais através da exploração assistida do código fonte do sistema legado; e (iii) Mineração de Aspectos, que identifica candidatos a interesses transversais a partir da mineração do código fonte do sistema legado.

A mineração de aspectos visa a identificar automaticamente os potenciais interesses transversais nos sistemas legados orientados a objetos que não utilizam programação orientada a aspectos. O processo de mineração de aspectos, de maneira geral, tenta

encontrar os interesses transversais candidatos a aspectos através da análise estática do código fonte por sintomas de espalhamento ou entrelaçamento de código, em um processo que tem por entrada o código fonte do sistema legado e como saída os candidatos a aspectos identificados.

Existem várias técnicas de mineração de aspectos propostas na literatura, das quais podem-se citar: a análise de *fan-in* [43], que identifica métodos que são invocados de vários locais; a detecção de clones [8] [32], que identifica métodos com código duplicado; teoria de grafos [58], que utiliza grafos que representam a conectividade entre os métodos; processamento de linguagem natural [63], que busca métodos relacionados semanticamente; análise de conceito formal [11] [67], que gera conceitos sobre o código analisado; análise de padrões [9] [10] [19] [53] [65] [70], que busca ocorrências de determinados padrões; aprendizado de máquina [61], que adquire conhecimento a partir de treinamento; e a análise de agrupamento (*clustering*) [12] [13] [27] [56] [57] [62], que usa algoritmos de agrupamento para formar grupos de candidatos a aspectos.

Dentre as técnicas citadas, a análise de agrupamento [69] destaca-se devido a característica de descobrir grupos naturais escondidos em um conjunto de elementos sem que seja necessário o conhecimento prévio de suas particularidades. A análise de agrupamento possui duas partes fundamentais: o algoritmo de agrupamento, que define a estratégia para o agrupamento dos elementos, e a medida de distância, que mensura a proximidade entre os elementos baseada na comparação de suas características.

A mineração de aspectos baseada em análise de agrupamento tem por característica a aplicação de algoritmos de agrupamento de forma a descobrir os grupos de métodos candidatos a aspectos que refletem os interesses transversais existentes em um sistema. Os trabalhos existentes na literatura aplicam os algoritmos de agrupamento: *k-means* [57], *k-medoids* [12], algoritmos genéticos (AG) [56], agrupamento hierárquico aglomerativo clássico [13] e CHAMELEON [27], enquanto as medidas de distância adotadas, de maneira geral, mensuram a proximidade entre os métodos pela similaridade dos nomes ou pela identificação do sintoma de espalhamento, relacionando os métodos que são chamados frequentemente de diferentes módulos.

O processo básico da mineração de aspectos baseada em agrupamento é composto tipicamente por quatro fases: Computação, onde o código fonte do sistema é analisado a fim de computar todas as classes, métodos e as relações de invocação entre eles; Filtro, onde são eliminados métodos e invocações que devem ser ignorados; Agrupamento, onde o algoritmo de agrupamento é aplicado para formar os grupos de candidatos a aspectos; e Análise, onde um analista valida os candidatos a aspectos e define a estratégia de refatoração. Este processo tem por saída os grupos de métodos candidatos a aspectos que representam a parte adendo do aspecto, que, para seu completo funcionamento, também exige a identificação dos pontos de corte. Somente o trabalho [27] explora a identificação de candidatos a pontos de corte, porém é limitado a categorização em execução ou chamada.

## 1.2 Motivação

Entende-se que a tarefa de mineração de aspectos em código fonte envolve a exploração de diferentes características e sintomas presentes no código fonte que apontam para a existência de interesses transversais que devem, na medida do possível, serem refatorados para aspectos. Entretanto, os trabalhos existentes na mineração de aspectos baseada em agrupamento não consideram todas as características conhecidas. Trabalhos existentes se apoiam na identificação de candidatos a aspectos baseada nos sintomas de espalhamento

e convenção de nomes de métodos. Características como código duplicado, convenção de nomes em nome de classes, padrões de retorno ou conjunto de parâmetros e uso de polimorfismo ainda não foram exploradas. Os trabalhos também não fazem o uso de combinação de todas estas características. Combinar as características pode levar a resultados melhores.

Nota-se que as técnicas existentes na mineração de aspectos baseada em agrupamento geram grupos de todos os métodos do sistema, sejam estes participantes de interesses transversais ou base, sem a preocupação da definição de instrumentos que permitam a eliminação dos interesses base ou a rápida separação dos potenciais interesses transversais dos interesses base. A criação destes tipos de instrumentos pode levar a uma melhor eficiência na mineração de aspectos.

Percebe-se também que as técnicas existentes se preocupam principalmente com a identificação de candidatos a aspectos que refletem a parte adendo, porém, um aspecto é formado por adendos e pontos de corte. Portanto, considera-se que é imprescindível a identificação dos pontos de corte que serão aplicados aos adendos identificados para a implementação dos aspectos. A criação de uma abordagem que integre a identificação dos principais tipos de pontos de corte com a identificação dos candidatos a aspectos (adendos) pode levar a uma melhor eficácia na mineração de aspectos.

### 1.3 Objetivos

Este trabalho tem como objetivo principal contribuir para a área de mineração de aspectos em código fonte baseada em análise de agrupamento através da proposição de uma nova abordagem integrada, fruto da pesquisa das melhores características encontradas em trabalhos existentes na literatura de mineração de aspectos. A abordagem proposta, chamada CAAMPI (*Clustering Based Approach for Aspect Mining and PointCut Identification*), contempla a identificação dos candidatos a aspectos através da definição de uma medida de distância que combina os sintomas de espalhamento, código duplicado e convenção de nomes. Além disto, a CAAMPI permite a identificação dos principais tipos de pontos de corte através da exploração de regras de associação.

A abordagem também inclui uma fase de pré-processamento que utiliza a técnica de *fan-in* para a eliminação parcial dos interesses base antes mesmo da fase de agrupamento, e uma fase de pós-processamento para a ordenação dos grupos obtidos na fase de agrupamento através de uma pontuação que indica o nível de potencial do grupo participar de um interesse transversal.

De forma a viabilizar a aplicação prática da abordagem, é proposto o *framework* CAAMPI4J, que define as características técnicas necessárias para permitir a criação de uma ferramenta automatizada que possibilita a análise de código fonte de sistemas desenvolvidos em linguagem Java, e, que também objetiva a servir de guia para a criação de ferramentas para outras linguagens.

O *framework* e a abordagem foram utilizados em experimentos com três sistemas reais, nos quais foram avaliadas diferentes combinações de medidas e tipos de algoritmos de agrupamento.

## 1.4 Organização

Este capítulo de introdução apresentou o contexto no qual este trabalho está inserido, a motivação para a sua realização e os objetivos. Os Capítulos 2, 3 e 4 são destinados a fundamentação teórica necessária para o entendimento deste trabalho. O Capítulo 2 apresenta os conceitos básicos de orientação a aspectos. O Capítulo 3 apresenta os conceitos básicos da análise de agrupamento e os tipos de técnicas existentes, detalhando os algoritmos utilizados neste trabalho. O Capítulo 4 apresenta os conceitos básicos de mineração de aspectos, aprofundando-se na aplicação da análise de agrupamento em código fonte e no detalhamento dos trabalhos relacionados. O Capítulo 5 apresenta a abordagem CAAMPI, detalhando seu processo, fases, medidas e algoritmos propostos, e os aspectos de implementação do *framework* CAAMPI4J. No Capítulo 6 é apresentada a metodologia aplicada nos experimentos, incluindo os índices de avaliação propostos, os resultados coletados e suas respectivas análises. Ao final, no Capítulo 7, são apresentadas as conclusões deste trabalho e as oportunidades identificadas para guiar possíveis trabalhos futuros.

O trabalho também conta com quatro apêndices. No Apêndice A são divulgadas as instâncias de interesses transversais consideradas nos experimentos. No Apêndice B são divulgados os resultados da etapa de calibração dos algoritmos, destacando as melhores configurações. No Apêndice C são listados todos os grupos obtidos pela execução da melhor instância obtida nos resultados para cada sistema, expondo os métodos de cada grupo e o valor obtido em cada uma das medidas de ordenação. No Apêndice D são listados os candidatos a pontos de corte obtidos para cada sistema vinculados a seus respectivos candidatos a aspectos.

## CAPÍTULO 2

# ORIENTAÇÃO A ASPECTOS

Neste capítulo são apresentados os conceitos básicos de orientação a aspectos que servirão como fundamento para o entendimento deste trabalho.

O capítulo está organizado nas seguintes seções: na Seção 2.1 é efetuada uma contextualização das limitações da orientação a objetos, na Seção 2.2 são apresentadas as vantagens da adoção da orientação a aspectos em sistemas orientados a objetos, na Seção 2.3 é apresentada uma visão geral do funcionamento das linguagens orientadas a aspectos, e na Seção 2.4 é introduzida a linguagem AspectJ.

### 2.1 Limitações da orientação a objetos

A utilização do paradigma orientado a objetos no desenvolvimento de software trouxe grandes evoluções quando comparado com a utilização de programação estruturada. Com o paradigma orientado a objetos, o sistema deixou de ser modelado através de funções procedurais e passou a ser modelado através de classes e objetos que encapsulam dados e comportamentos que representam a solução de um problema de maneira simples e prática.

Apesar da utilização da orientação a objetos proporcionar que o software seja modelado visando alta coesão e baixo acoplamento, existem algumas situações que limitam a eficácia de uma boa modelagem do software que aplica unicamente este paradigma.

Conforme o software foi aumentando em complexidade, foram sendo criados guias e conceitos de boas práticas de modelagem, chamados de Padrões de Projeto (do inglês *Design Patterns*) [4]. Uma prática amplamente adotada foi a divisão em camadas, especialmente o modelo MVC [66] (Modelo-Visão-Controle, do inglês *Model-View-Controller*). Esta prática traz grandes benefícios na modelagem dos interesses base (do inglês *core concerns*) já que divide o software em camadas que apresentam funcionalidades e comportamentos próprios, auxiliando na redução da complexidade. Porém, existem algumas funcionalidades que não fazem parte dos interesses base e que necessitam cruzar estas camadas para o seu total funcionamento, estes são denominados interesses transversais (do inglês *crosscutting concerns*).

Um interesse transversal é então definido como “uma entidade independente que cruza transversalmente outras funcionalidades de um software” [15], portanto, pode-se definir que os interesses transversais são funcionalidades que, para seu completo e eficaz funcionamento, necessitam da troca de mensagens e da realização de operações entre diversos componentes de distintos módulos dentro de um software.

Exemplos de interesses transversais incluem: controle de acesso, controle de transações, geração de *logs*, tratamento de exceções, *cache*, controle de concorrência, garantia de integridade de transações, monitoração de desempenho, *pool* de recursos, persistência de dados, armazenamento, entre outros. Todos os exemplos citados cortam várias camadas do sistema conforme ilustrado na Figura 2.1.

Devido a natureza dos interesses transversais cortarem mais de uma camada, o projeto de uma aplicação OO está sujeito a dois problemas comuns conforme [15]:

**Espalhamento de código** (do inglês *code scattering*): Ocorre quando uma funcionalidade é implementada de forma espalhada em vários módulos. Existem dois tipos



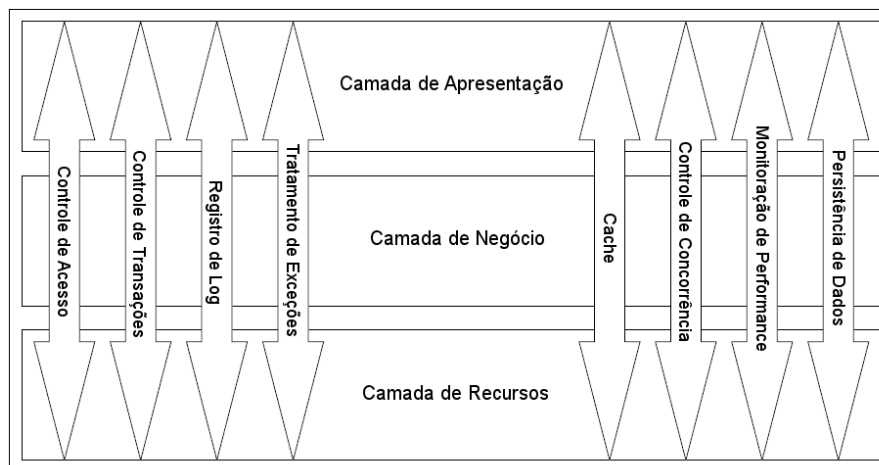


Figura 2.1: Interesses transversais cruzam várias camadas de um software (adaptada de [15]).

principais de consequências no projeto:

- Blocos de código duplicado: As operações de uma funcionalidade são duplicadas (Clonadas) em vários módulos. Ex: log, tratamento de exceções. Exemplificado na Figura 2.2.

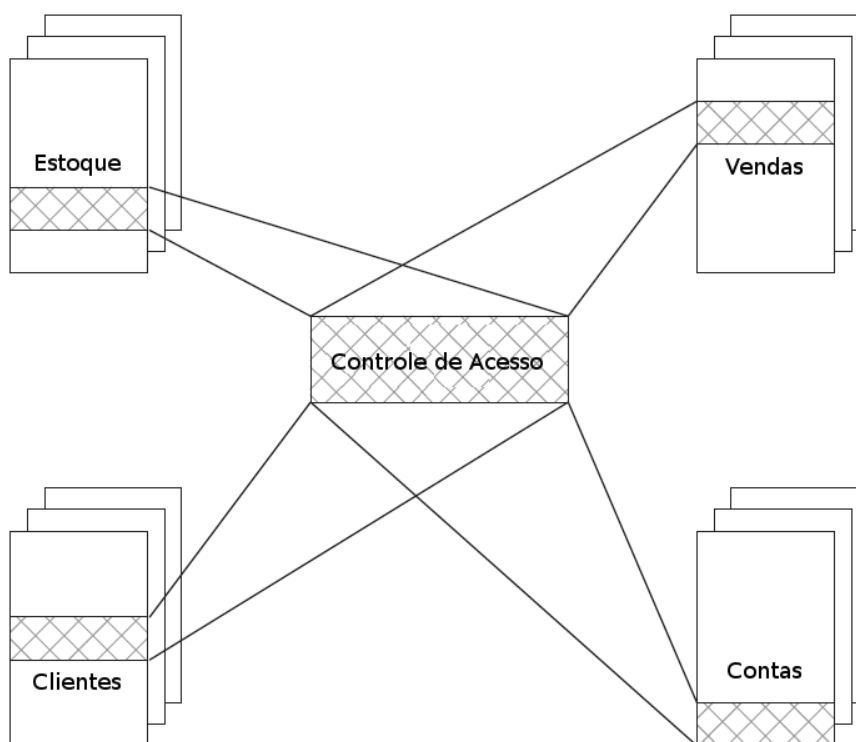


Figura 2.2: Espalhamento de código devido a código duplicado (adaptada de [39]).

- Blocos de código complementar: Diferentes módulos implementam partes complementares de uma determinada funcionalidade. Ex: controle de acesso, um módulo

implementa a autenticação, outro módulo implementa a autorização. Exemplificado na Figura 2.3.

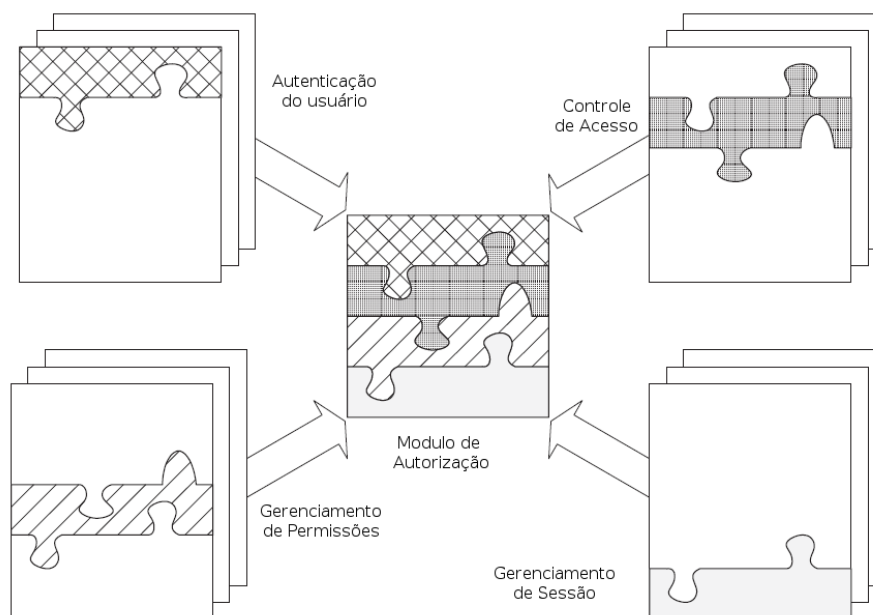


Figura 2.3: Espalhamento de código devido a código complementar (adaptada de [39]).

A Figura 2.4 exibe um exemplo prático de espalhamento de código na implementação do software Apache Tomcat 4. As colunas representam os módulos do sistema e as linhas em destaque representam as operações da funcionalidade de log.

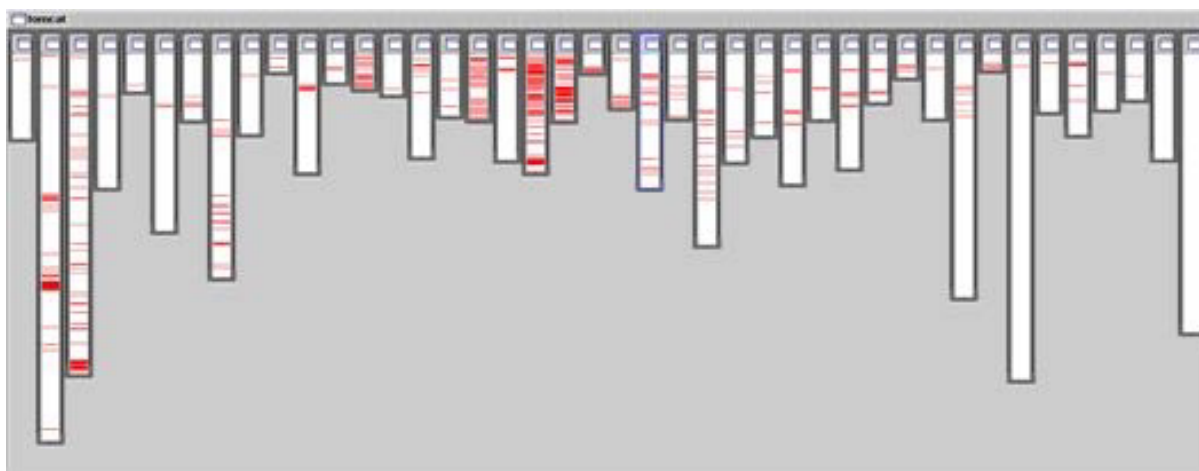


Figura 2.4: Funcionalidade de log espalhada por vários módulos no Apache Tomcat 4 (extraída de [15]).

**Código entrelaçado** (do inglês *code tangling*): Ocorre quando um módulo tem que gerenciar várias funcionalidades dentro de um mesmo contexto ou quando o módulo possui elementos de implementação de outras funcionalidades dentro dele. O tipo principal

de consequência no projeto do software são os blocos de códigos com múltiplas funcionalidades, ou seja, que efetuam operações além do contexto principal do módulo, o que reduz a coesão e aumenta o acoplamento entre módulos. Ex: Método que tem por objetivo principal o cadastro de clientes, porém, além de seu objetivo, faz o controle de acesso, o log e a persistência dos dados. Exemplificado na Figura 2.5

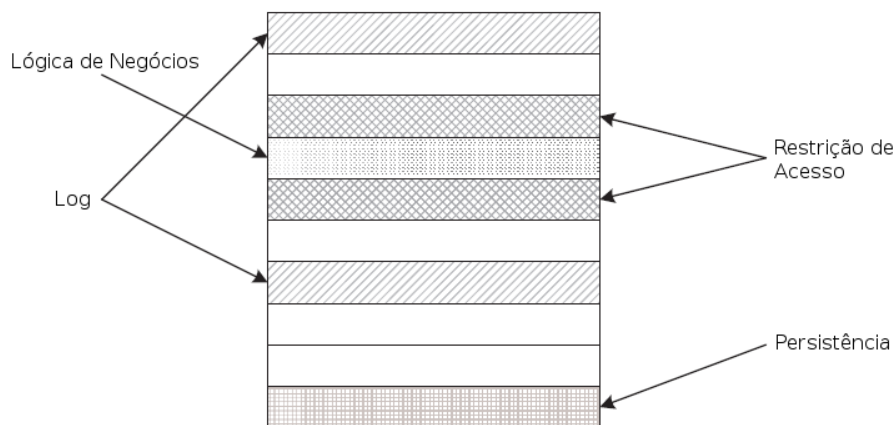


Figura 2.5: Módulo com sintoma de código entrelaçado (adaptada de [39]).

De acordo com [39], os principais impactos no processo de desenvolvimento de software causados pelo problema de código espalhado são:

- **Produtividade baixa:** As implementações das funcionalidades que entrecortam vários módulos forçam o desenvolvedor a mover o foco de solução para todos os módulos que necessitam daquela funcionalidade, exigindo controle extra para rastrear os códigos que estão espalhados e duplicação desnecessária de código que efetua as mesmas operações.
- **Rastreabilidade:** Como uma funcionalidade é implementada em vários pontos, torna-se difícil rastrear todos os pontos onde ela está presente, e no caso de evoluções, manutenções ou resolução de problemas, é necessário checar todos os módulos nos quais a funcionalidade foi implementada.

Já em relação ao problema de código entrelaçado, os principais impactos causados no processo de desenvolvimento de software são:

- **Dificuldade de evolução:** Módulos têm sua implementação acoplada com outras funcionalidades, dificultando a sua manutenção, evolução e a identificação de erros.
- **Redução de qualidade:** Módulos não possuem funcionalidades principais claras, pois efetuam várias operações referentes a outras funcionalidades que não fazem parte de seu contexto principal.
- **Código não reutilizável:** Se uma implementação envolve várias funcionalidades que não fazem parte do contexto principal, ela dificilmente será útil para reaproveitamento em outros cenários.

Como é possível observar, a Orientação a Objetos não é suficiente para modelar um software com total separação de interesses. Devido a este fato foi necessária a pesquisa de novas alternativas para reduzir estas limitações.

## 2.2 Reduzindo limitações com a Orientação a Aspectos

Tendo em vista a existência dos problemas apresentados que podem ocorrer no desenvolvimento de software orientado a objetos, surgiu a partir do estudo de Kiczales [37] a programação orientada a aspectos (OA). A OA é uma alternativa de implementação dos interesses transversais que evita o acoplamento entre os módulos através da criação dos chamados aspectos. Os aspectos colaboram com as classes e eliminam o envolvimento das classes que implementam interesses base nas implementações dos interesses transversais. Sendo assim, as classes podem evoluir livremente sem que haja preocupação da dependência gerada com os interesses transversais. Para que esta interação ocorra, as funcionalidades dos interesses transversais implementadas por meio de aspectos são integradas às classes utilizando um combinador de aspecto (do inglês *aspect weaver*), que combina o aspecto e o objeto através de um compilador especial, ou de uma classe procuradora (do inglês *proxy class*), que modifica a classe em tempo de execução para incluir as funcionalidades introduzidas pelos aspectos e passa a então representar a classe original.

Os benefícios da utilização de aspectos são vários, pode-se enumerar alguns deles conforme [39]:

- Módulos com responsabilidades claras: Aspectos permitem que os módulos tenham responsabilidades somente sobre suas funcionalidades principais. Isto resulta numa clara definição de responsabilidades, além de aprimorar o rastreamento de funcionalidades.
- Alta modularização: Aspectos proveem mecanismos para definir cada funcionalidade separadamente com o mínimo de acoplamento. Isto resulta em uma implementação modularizada mesmo quando interesses transversais estão presentes, reduzindo a duplicação de código e tornando o sistema mais fácil de ser entendido e de ser mantido.
- Facilidade para evolução do sistema: Aspectos são modularizados individualmente e tornam os interesses base independentes dos interesses transversais. Adicionar novos interesses transversais não requer alterações nos interesses base, e, no caso de adicionar novos interesses base, os aspectos existentes podem simplesmente entrecortar (do inglês *crosscutting*) as novas funcionalidades. Isto resulta em uma rápida implementação de novos requisitos.
- Decisões de projeto podem ser postergadas: Com aspectos o arquiteto não precisa se preocupar tanto com decisões de requisitos futuros, já que novos requisitos que caracterizam interesses transversais podem ser implementados com a criação de novos aspectos, sem interferir na implementação dos interesses base existentes.
- Maior reúso de código: Aspectos são implementados como módulos individuais, e cada módulo possui menor acoplamento de que seus equivalentes em implementações orientadas a objetos. Como os interesses base desconhecem os acoplamentos gerados pelos interesses transversais, já que os aspectos são integrados posteriormente pelo combinador, o reaproveitamento de implementações é facilitado.
- Tempo para entrega reduzido: Decisões de projeto tornam-se mais ágeis devido a possibilidade de postergar preocupações com novos requisitos. A clara separação

de responsabilidades aumenta a possibilidade de paralelismo de atividades, aumentando a produtividade. Mais reuso de código reduz o tempo de desenvolvimento. Evolução fácil permite uma resposta rápida para atendimento de novos requisitos. Todas estas características levam a sistemas que são mais rápidos de desenvolver, propiciando entregas mais rápidas.

- **Redução de custos:** Como os aspectos evitam os custos de modificação dos interesses base que implementam interesses transversais, eles tornam mais barata a implementação das funcionalidades de interesses transversais. Além disso, como os desenvolvedores podem focar na implementação dos interesses base sem se preocupar com os interesses transversais, eles podem efetuar atividades mais voltadas a suas especialidades, reduzindo o custo de implementação dos interesses base.

Os benefícios que os aspectos trazem ao desenvolvimento de software fazem da Orientação a Aspectos uma opção atraente para a melhoria de todo o processo de desenvolvimento. Para que seja possível aproveitar o potencial dos aspectos, é necessária a utilização de uma linguagem Orientada a Aspectos.

## 2.3 Linguagens orientadas a aspectos

Basicamente uma linguagem orientada a aspectos é composta de duas partes [39]:

**Especificação da linguagem:** Descreve a construção da linguagem e a sintaxe que será utilizada. Na especificação de uma linguagem orientada a aspectos é necessária a utilização de duas linguagens: a linguagem nativa orientada a objetos para a implementação dos interesses e a linguagem utilizada para implementação das regras do combinador. A linguagem para implementação das regras do combinador especifica como os interesses implementados na linguagem nativa serão integrados para gerar o sistema final. Esta linguagem pode ser uma extensão da linguagem natural ou uma linguagem totalmente diferente.

**Implementação da linguagem:** Verifica a aderência do código à especificação da linguagem e traduz o código para a forma executável. A implementação da linguagem ocorre em dois passos: A combinação, que define quais e como os módulos do sistema serão combinados conforme as regras de combinação especificadas, e o combinador, que gera o novo código fonte do sistema com o código dos interesses base e dos aspectos combinados para que seja processado pelo compilador ou interpretador.

Atualmente existem algumas implementações de linguagens orientadas a aspectos dentre as quais destacam-se: AspectC++ [64] para C++, Pythius [52] para Python, JAC (*Java Aspect Component*) [5] e AspectJ [36] [39] para Java. Na seção a seguir são descritos aspectos de implementação da linguagem AspectJ, uma das mais utilizadas na literatura.

## 2.4 Introdução a Linguagem AspectJ

AspectJ [36] [39] é uma linguagem orientada a aspectos que funciona como uma extensão a linguagem Java. Os aspectos são implementados utilizando linguagem Java e o compilador do AspectJ gera como saída arquivos de classe (.class) compatíveis para execução na própria máquina virtual Java.

Para a implementação das regras de combinação em AspectJ é utilizado o entrecorte. Um entrecorte pode definir regras de combinação que podem cortar múltiplos módulos

do sistema de maneira sistemática de forma a modularizar os interesses transversais. São definidos dois tipos de entrecortes [39]:

- **Dinâmico:** Combina novos comportamentos durante a execução de um programa. O fluxo de execução do programa é ampliado e substituído através de entrecortes nos módulos, modificando o comportamento original dos módulos do sistema.
- **Estático:** Introduce modificações na estrutura das classes, interfaces e aspectos do sistema. Este tipo aplica alterações estruturais que não modificam o comportamento interno dos módulos afetados.

O AspectJ utiliza extensões da linguagem Java para especificar as regras de combinação tanto para entrecorte dinâmico quanto estático, são elas [39]:

### Ponto de Junção (do inglês *Join Point*)

É um ponto identificável na execução de um programa onde podem ser injetados os entrecortes. São exemplos de ponto de junção as chamadas de método, a execução de método, a instanciação de um objeto, o acesso a um atributo e o tratamento de exceções. No exemplo abaixo podem ser identificados a execução do método `deposito()` e o acesso ao atributo `saldo`.

```
public class Conta {
    float saldo = 0.0f;
    void deposito(float valor) {
        saldo += valor;
    }
}
```

### Ponto de Corte (do inglês *Pointcut*)

São elementos que instruem como os pontos de junção serão selecionados. Os pontos de corte podem capturar ou identificar pontos de junção no fluxo de programa, além de possibilitar expor o seu contexto. No exemplo abaixo é exibido como capturar a execução do método `deposito()` da classe `Conta`.

```
execution(void Conta.deposito(float))
```

### Adendo (do inglês *Advice*)

Um adendo é uma construção que provê uma maneira de expressar a ação de um entrecorte em um ponto de junção que foi capturado por um ponto de corte. O corpo de um adendo é similar a um corpo de método já que encapsula a lógica que será executada quando for atingido o ponto de junção. Há três tipos de adendos:

- **Adendo antes (*Before*):** Executa a ação do adendo antes da execução original do ponto de junção.
- **Adendo após (*After*):** Executa a ação do adendo após a execução original do ponto de junção.
- **Adendo durante (*Around*):** É um tipo especial de adendo que executa a ação do adendo e altera a execução de um ponto de junção, podendo então, ignorar a execução, continuar a execução original ou executar o ponto de junção com o contexto alterado.

No exemplo abaixo é criado um adendo para execução antes do método `deposito()` da classe `Conta`.

```
before() : execution(void Conta.deposito(float)) {
    System.out.println("Executando depósito");
}
```

Juntos, pontos de cortes e adendos formam as regras de entrecorte dinâmico. Enquanto os pontos de corte identificam os pontos de junção requeridos, os adendos determinam as ações que serão executadas quando ocorrerem os pontos de junção.

### Introdução (do inglês *Introduction*)

É uma instrução, também conhecida como declaração inter tipos, de entrecorte estático que introduz alterações nas classes, interfaces e aspectos do sistema. É possível introduzir métodos, atributos ou modificar a hierarquia de classes. Na declaração abaixo é alterada a hierarquia de classes da classe Conta através da implementação da interface Deposito:

```
declare parents: Conta implements Deposito;
```

### Declaração em tempo de compilação (do inglês *Compile-time declaration*)

São declarações de entrecorte estático que não alteram o comportamento do sistema. Elas permitem adicionar novos tipos de advertências (*warning*) e erros (*error*) após detectar certos padrões, fazendo com que o compilador exiba advertências na sua saída ou aborte a compilação caso encontre o tipo de erro declarado. O exemplo abaixo apresenta a declaração de uma advertência que será lançada caso qualquer método do sistema chame o método credito() da classe Conta.

```
declare warning : call(void Conta.credito(Object))
    : "Método obsoleto, utilize o método Conta.deposito()";
```

### Relaxamento de exceção (do inglês *Exception softening*)

Permite que uma exceção com tratamento obrigatório (*Checked Exception*) seja relaxada e tratada como uma exceção que não necessita de tratamento obrigatório (*Unchecked Exception*). O relaxamento de exceção evita o entrelaçamento de código já que evita o tratamento de exceções obrigatórias na lógica de implementação dos interesses. No exemplo a seguir o método executar() da classe TesteRemoto terá a exceção RemoteException, que é uma exceção de tratamento obrigatório, relaxada, ou seja, o método executar() não necessitará tratar e nem lançar a exceção RemoteException.

```
declare soft : RemoteException : call(void TesteRemoto.executar());
```

### Aspecto

O aspecto é a unidade central do AspectJ. Ele contém o código que expressa as regras de combinação tanto para entrecorte dinâmico quanto estático. São combinados em um aspecto os pontos de corte, adendos, introduções e declarações. Aspectos também podem conter atributos, métodos e classes aninhadas. No exemplo abaixo é demonstrado como os exemplos anteriores são encapsulados em um aspecto.

```
public aspect AspectoExemplo {
    before() : execution(void Conta.deposito(float)) {
        System.out.println("Executando depósito");
    }

    declare parents: Conta implements Deposito;
```

```
declare warning : call(void Conta.credito(Object))
    : "Método obsoleto, utilize o método Conta.deposito()";

declare soft : RemoteException : call(void TesteRemoto.executar());
}
```

Resumidamente, para encapsular um interesse transversal utilizando as extensões do AspectJ é necessário [39]:

1. Criar um aspecto que irá encapsular o interesse transversal;
2. Declarar atributos, métodos ou classes aninhadas que sejam necessários;
3. Criar os pontos de corte que irão capturar os pontos de junção desejados;
4. Criar os adendos para os pontos de corte contendo o corpo dos métodos;
5. Criar declarações ou introduções que sejam necessárias.

## 2.5 Considerações Finais

Neste capítulo foram apresentados os principais conceitos de Orientação a Aspectos, enfatizando os problemas existentes na Orientação a Objetos que motivaram a criação da Orientação a Aspectos, as vantagens da utilização da Orientação a Aspectos, o funcionamento básico de uma linguagem orientada a aspectos e uma pequena introdução da linguagem AspectJ.

Este capítulo é a primeira parte dos fundamentos teóricos deste trabalho. O próximo capítulo é destinado a apresentação dos conceitos básicos da técnica de agrupamento.



## CAPÍTULO 3

### ANÁLISE DE AGRUPAMENTO

Neste capítulo são apresentados os conceitos básicos da técnica de análise de agrupamento (*Clustering*) que servem de fundamento para o entendimento deste trabalho.

O capítulo está organizado nas seguintes seções: na Seção 3.1 é feita uma contextualização da análise de agrupamento e de seu processo básico, na Seção 3.2 são apresentadas as categorias de medidas de distância que podem ser utilizadas com os algoritmos de agrupamento e na Seção 3.3 são apresentados os tipos de algoritmos de agrupamentos existentes.

#### 3.1 Introdução a Análise de Agrupamento

A sociedade vive em uma era na qual estão disponíveis muitos dados advindos de vários tipos de medidas e observações. Estes dados definem propriedades, características, resultados, estados, entre outras possibilidades. Para que a partir destes dados seja possível identificar comportamentos, efetuar classificações, categorizações ou agrupamento de semelhantes em conjuntos, tem-se como opção agrupá-los através da análise de suas propriedades similares ou algum critério específico.

A maneira como os dados serão categorizados depende do conhecimento prévio das classes em que os dados serão classificados, existindo para tanto dois tipos de sistemas de classificação:

- **Supervisionado:** Na qual os dados de entrada não rotulados, ou seja, que possuem naturezas desconhecidas da qual se tem pouca ou nenhuma informação prévia, são classificados em um conjunto finito e discreto de classes previamente conhecidas.
- **Não Supervisionado:** Também chamado de agrupamento (do inglês *clustering*) ou análise exploratória de dados. Na qual os dados de entrada não rotulados são classificados em um conjunto finito e discreto de estruturas de dados desconhecidas que possuem características similares.

A técnica de agrupamento é do tipo não supervisionada, portanto é esperado que a mesma descubra os grupos naturais que existem em um conjunto de dados sem que seja necessário o conhecimento prévio de suas características, ou seja, que busque padrões escondidos nos próprios dados.

Uma forma de representar o problema de agrupamento é através da identificação de cada dado como um ponto em um espaço multi-dimensional de características. Sendo assim, a entrada do algoritmo de agrupamento é representada por um conjunto de pontos neste espaço de características e a sua saída é a categorização desses dados em classes, ou seja, o possível grupo ao qual cada dado deve pertencer [14]. Este processo é denominado análise de agrupamento.

##### 3.1.1 Processo de Análise de Agrupamento

O processo de análise de agrupamento, segundo [69], é composto de quatro passos básicos conforme ilustrado na Figura 3.1:

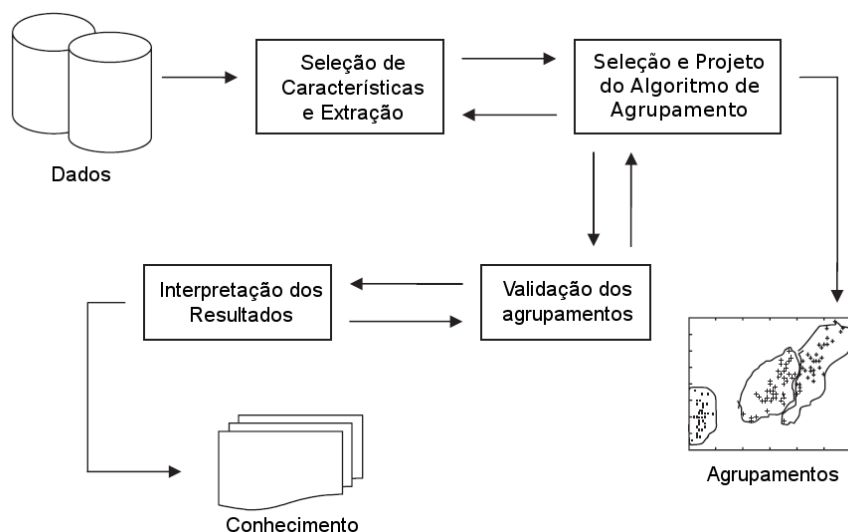


Figura 3.1: Processo de Análise de Agrupamento (adaptada de [69]).

### 1. Seleção de características e extração

A seleção de características visa a escolher características determinantes de um conjunto de características candidatas que sejam eficazes na distinção dos objetos, enquanto a extração aplica transformações nos dados para modificar as características originais em características mais eficazes para efetuar a distinção dos objetos. Uma boa seleção ou transformação reduz a necessidade de armazenamento de dados, o custo da medida de similaridade, simplifica o projeto e facilita o entendimento dos dados. Em um cenário ideal devem ser selecionadas características com padrões de distinção imunes a ruídos e fáceis de se obter e interpretar.

### 2. Seleção e projeto do algoritmo de agrupamento

Consiste na escolha de uma medida de distância apropriada, de um algoritmo de agrupamento e definição das restrições de agrupamento. Uma vez que estes itens sejam determinados, o algoritmo de agrupamento pode ser modelado como um problema de otimização. Os tipos de medidas de distância são apresentados na Seção 3.2 e tipos de algoritmos de agrupamento são apresentados na Seção 3.3.

### 3. Validação dos agrupamentos

Consiste na aplicação de índices de forma a avaliar padrões efetivos e critérios de extrema importância com o intuito de definir um grau de confiança dos resultados do algoritmo de agrupamento. Há três classificações para índices [69]:

- Externo: É baseado no conhecimento prévio da classificação dos dados. A informação prévia é utilizada como um padrão para validar a solução de agrupamento.
- Interno: Não dependente de conhecimento prévio. Examina a estrutura de agrupamento diretamente a partir dos dados originais.
- Relativo: Enfatiza a comparação de diferentes estruturas de agrupamento de maneira a prover uma referência para decidir qual algoritmo melhor revela as características dos objetos.

#### 4. Interpretação dos resultados

Consiste na interpretação dos resultados para obtenção de informações úteis sobre os dados analisados. O intuito é chegar a um entendimento claro do que os resultados representam, para então efetivamente relatar que um conhecimento foi obtido a partir destes dados.

### 3.2 Medidas de Distância

As medidas de distância são utilizadas pelos algoritmos de agrupamento para mensurar a proximidade entre os elementos, baseando-se na comparação de suas características. As medidas devem ser cuidadosamente selecionadas, pois, estas são a base principal para a efetividade do algoritmo de agrupamento.

São comumente encontradas na literatura duas categorias de medidas [14]:

#### Dissimilaridade

Este tipo de medida mensura a dissimilaridade entre as características dos elementos, ou seja, ela aumenta proporcionalmente conforme os padrões de características diferem entre si.

A medida de dissimilaridade mais popular é a distância Euclidiana [14]. A distância Euclidiana entre dois vetores de dados  $\vec{Z}_u = \{z_{u,1}, \dots, z_{u,k}\}$  e  $\vec{Z}_v = \{z_{v,1}, \dots, z_{v,k}\}$ , onde  $k$  é o número de dimensões (atributos), é dada na Equação 3.1.

$$d(\vec{Z}_u, \vec{Z}_v) = \sqrt{\sum_{i=1}^k (z_{u,i} - z_{v,i})^2} = \|\vec{z}_u - \vec{z}_v\| \quad (3.1)$$

#### Similaridade

Este tipo de medida mensura a similaridade entre as características dos elementos, ou seja, ela aumenta proporcionalmente conforme os padrões de características possuem similaridades entre si.

A medida de similaridade mais comum é a distância Cosine [14]. A distância Cosine entre dois vetores de dados  $\vec{Z}_u = \{z_{u,1}, z_{u,2}, \dots, z_{u,k}\}$  e  $\vec{Z}_v = \{z_{v,1}, z_{v,2}, \dots, z_{v,k}\}$ , onde  $k$  é o número de dimensões (atributos), é dada na Equação 3.2

$$d(\vec{Z}_u, \vec{Z}_v) = \frac{\sum_{i=1}^k z_{u,i} \times z_{v,i}}{\sqrt{\sum_{i=1}^k (z_{u,i})^2} \times \sqrt{\sum_{i=1}^k (z_{v,i})^2}} \quad (3.2)$$

### 3.3 Tipos de Algoritmos de Agrupamento

Vários algoritmos de agrupamento foram desenvolvidos para resolver diferentes problemas de uma grande variedades de campos, porém, não existe um algoritmo de propósito geral que resolva todos os problemas. Sendo assim, é importante investigar cuidadosamente as características do problema que será resolvido de forma a escolher a estratégia de agrupamento adequada [69].

Em geral, os algoritmos de agrupamento podem ser classificados em 5 categorias [25]: Particionamento, Hierárquico, Baseado em densidade, Baseado em grade e Baseado em modelo.

### Particionamento

Os algoritmos baseados em particionamento criam grupos a partir da distribuição dos elementos (dados) em partições. Estes algoritmos utilizam uma técnica de otimização iterativa de realocação que tenta melhorar a partição através da movimentação dos elementos de um grupo para outro.

Cada partição deve conter ao menos um elemento e cada elemento deve pertencer a exatamente uma partição. O critério geral para uma boa partição é que elementos do mesmo grupo devem ser os mais homogêneos possíveis, enquanto elementos de diferentes grupos devem ser os mais heterogêneos possíveis. Para que se possa atingir o ótimo global, o algoritmo requer uma enumeração exaustiva das possíveis partições de elementos.

Os algoritmos de otimização clássicos aplicados a particionamento são:

- *k-means*: O *k-means* representa cada grupo por um elemento fictício, denominado centro do grupo (do inglês *cluster center*) ou centróide. O algoritmo iterativamente associa cada elemento ao centro de grupo mais próximo e recalcula cada centro de grupo com a média dos atributos dos elementos associados. A heurística do *k-means* é esboçada no Algoritmo 1.

<p><b>Entrada:</b> <math>n</math>: número de partições  <b>Saída:</b> <math>K = \{K_1, \dots, K_n\}</math>: partições</p> <pre> 1 início 2   <math>f_1, \dots, f_n</math>: centros das partições <math>K_1, \dots, K_n</math>; 3   para <math>j \leftarrow 1</math> até <math>n</math> faça 4     <math>f_j \leftarrow \text{aleatorio}()</math>; // Inicializa centro com valor aleatório 5   fim 6   enquanto Critério de parada não atingido faça 7     para <math>j \leftarrow 1</math> até <math>n</math> faça 8       // Associa cada elemento com o centro mais próximo 9       <math>K_j \leftarrow \{e_i   \forall 1 \leq r \leq n, r \neq j, d(e_i, f_j) \leq d(e_i, f_r)\}</math>; 10      <math>f_j \leftarrow</math> A média dos objetos em <math>K_j</math>; 11     fim 12 fim</pre>
--

**Algoritmo 1:** *k-means*.

- *k-medoids*: O *k-medoids* representa cada grupo por um de seus elementos que está localizado o mais próximo ao centro do grupo. A este elemento central é dado o nome de *medoid*. O algoritmo iterativamente associa cada elemento ao *medoid* mais próximo, e, ao final, o algoritmo troca os *medoids* por outros elementos não *medoids* e seleciona a configuração de menor custo. A heurística do *k-medoids* é esboçada no Algoritmo 2. Este algoritmo também é chamado de PAM (*Partitioning Around Medoids*).

<p><b>Entrada:</b> <math>n</math>: número de partições  <b>Saída:</b> <math>K = \{K_1, \dots, K_n\}</math>: partições</p> <pre> 1 início 2   Seleccione <math>x</math> dos <math>n</math> elementos como medoids ; 3   enquanto Critério de parada não atingido faça 4     Associe cada elemento <math>e</math> ao seu medoid mais próximo ; 5     para cada medoid <math>m</math> faça 6       para cada não medoid <math>o</math> faça 7         Troque <math>m</math> e <math>o</math> e compute o custo da nova configuração ; 8       fim 9     fim 10    Seleccione a configuração com o custo mais baixo ; 11  fim 12 fim </pre>
---

**Algoritmo 2:** *k-medoids*.

Além dos métodos de otimização clássicos apresentados, também podem ser aplicados na busca da melhor partição de elementos outros métodos de otimização baseados em Inteligência Artificial (IA), como Algoritmos Genéticos [20], Otimização por Nuvem de Partículas [35], entre outros.

### Hierárquico

Os algoritmos baseados no método hierárquico criam uma decomposição hierárquica do conjunto de dados. Existem dois métodos para a formação da hierarquia [69]:

- Aglomerativo: É baseado na maneira como uma decomposição hierárquica é formada. O algoritmo inicia com cada elemento pertencendo a um grupo individual, as folhas, e sucessivamente junta os grupos que estão próximos até que os grupos estejam agrupados em um único grupo, o topo da hierarquia.
- Divisivo: O algoritmo inicia com todos os elementos pertencendo a um único grupo, o topo da hierarquia. A cada iteração um grupo é dividido em grupos menores até que todos os elementos possuam o seu grupo individual, as folhas.

Ambos os métodos organizam os dados em uma estrutura hierárquica baseada em uma matriz de distância. Os resultados podem ser representados em uma árvore binária conforme exemplificado na Figura 3.2.

O nó principal, o topo da hierarquia, representa todo o conjunto de elementos, e cada folha representa um elemento. Já os nós intermediários descrevem o quanto os elementos estão próximos entre si.

O método aglomerativo é o mais utilizado, pois, quando comparado com método divisivo, este necessita de menos recursos computacionais.

O algoritmo aglomerativo clássico é composto dos seguintes passos:

1. Cada dado inicia pertencendo a seu próprio grupo, formando  $N$  grupos.
2. Calcula a matriz de distância (através da medida de distância de ligação entre grupos escolhida) para os  $N$  grupos.

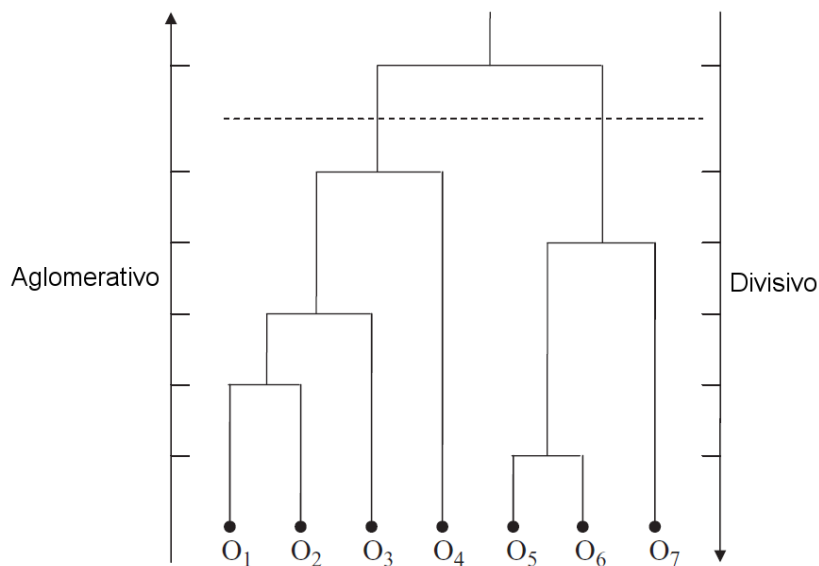


Figura 3.2: Agrupamento em estrutura hierárquica (adaptada de [69]).

3. Busca a distância mínima entre os grupos.  $D(C_i, C_j) = \min \{D(C_m, C_l)\}$ , onde  $D(C_m, C_l)$  é a medida de distância de ligação e  $1 \leq m, l \leq N$ ,  $m \neq l$ . Combina os grupos  $C_i$  e  $C_j$ , formando o novo grupo  $C_{ij}$ .
4. Atualiza a matriz de distância através da computação das distâncias entre o grupo  $C_{ij}$  e os demais grupos.
5. Repete os Passos 3 e 4 até que reste somente um grupo.

Existem várias medidas de ligação entre grupos na literatura, dentre as quais pode-se citar [69]:

- *single linkage*: A distância entre um par de grupos é determinada pelo dois elementos mais próximos entre os grupos. Esta distância também é chamada de método do vizinho mais próximo.
- *complete linkage*: A distância entre um par de grupos é determinada pelos dois elementos mais distantes entre os grupos.
- *average linkage*: A distância entre um par de grupos é determinada pela média das distâncias entre todos os pares de elementos dos grupos.
- *weighted average linkage*: Calcula a média das distâncias entre todos os pares de elementos dos grupos como na medida *average linkage*, porém, a distância entre o grupo formado e o restante tem pesos definidos baseados no número de elementos em cada grupo.
- *centroid linkage*: Dois grupos são mesclados de acordo com a distância de seus centros de grupos (centróides).

Além do algoritmo clássico, existem outros algoritmos populares na literatura com heurísticas aperfeiçoadas, como o BIRCH (*Balanced Iterative Reducing and Clustering using hierarchies*) [73], o CURE (*Clustering Using Representatives*) [23], e o CHAMELEON [33].

O CHAMELEON [33] é um algoritmo de agrupamento hierárquico aglomerativo que utiliza um modelo dinâmico para identificação dos grupos e representa o espaço vetorial como um grafo em que os nós são os elementos e os vértices determinam a similaridade entre os atributos dos elementos. O processo do CHAMELEON é representado na Figura 3.3.

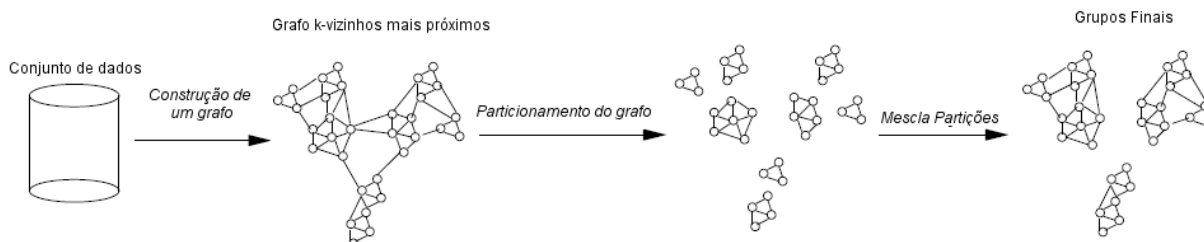


Figura 3.3: Processo do algoritmo CHAMELEON (adaptada de [33]).

O CHAMELEON primeiramente constrói um grafo com os  $k$ -vizinhos mais próximos, no qual, um vértice é mantido somente se ambos os vértices são  $k$ -vizinhos mais próximos. Os grupos são encontrados em duas fases: (i) Aplicação de um algoritmo de particionamento de grafos para agrupar os itens de dados em uma grande quantidade de sub-grupos relativamente pequenos, e (ii) Aplicação de um algoritmo de agrupamento hierárquico aglomerativo para encontrar os grupos genuínos através da repetida combinação destes sub-grupos.

A característica principal do CHAMELEON é que no processo de agrupamento, dois sub-grupos  $C_i$  e  $C_j$  são mesclados levando em conta tanto a interconectividade quanto a proximidade entre eles.

### Baseado em densidade

Os algoritmos baseados em densidade criam partições de grupos baseadas na noção de densidade. Diferentemente do particionamento, este continua a expandir os grupos até que a densidade de sua vizinhança atinja um certo limite estipulado. Este método pode ser utilizado para filtrar ruídos nos dados, ou descobrir grupos com formas arbitrárias.

O algoritmo de agrupamento baseado em densidade mais comum na literatura é o DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) [17].

### Baseado em grade

Os algoritmos baseados em grade quantificam objetos no espaço em um número finito de células que formam uma estrutura em grade. Todas as operações de agrupamento são executadas nesta estrutura de grade.

Exemplos de algoritmos baseados em grade são o STING (*STatistical INformation Grid*) [68], o WaveCluster [59] e o FC (*Fractal Clustering*) [7].

### Baseado em modelo

Os algoritmos baseados em modelo criam um modelo hipotético para cada um dos grupos e encontram a melhor classificação dos dados para determinado modelo. Este método pode localizar grupos através da construção de uma função de densidade

que reflete a distribuição espacial dos pontos de dados. Este também permite obter de forma automatizada o número de grupos utilizando uma base estatística.

Um exemplo de algoritmo baseado em modelo é o EM (*Expectation Maximization*) [44].

### 3.4 Considerações Finais

Neste capítulo foram apresentados os principais conceitos de análise de agrupamento. Foi efetuada uma contextualização sobre o processo de agrupamento e a apresentação das categorias de medidas de distância e dos tipos de algoritmos de agrupamento existentes na literatura.

Como mencionado, os diferentes algoritmos de agrupamento exploram distintas características, não existindo para tanto um algoritmo de propósito geral que atenda a todas as situações. Por isso é importante a exploração e comparação de vários algoritmos de agrupamento e medidas de distância para cada problema específico, tal como a mineração de aspectos, foco deste trabalho.

Este capítulo é a segunda parte dos fundamentos teóricos deste trabalho. O próximo capítulo é destinado a apresentação dos conceitos básicos da técnica de mineração de aspectos, o aprofundamento na aplicação da análise de agrupamento na mineração de aspectos e os trabalhos relacionados.



## CAPÍTULO 4

### MINERAÇÃO DE ASPECTOS

Neste capítulo são apresentados os conceitos básicos de mineração de aspectos, os trabalhos relacionados e o levantamento das necessidades e oportunidades que motivaram a proposta do presente trabalho.

O capítulo está organizado nas seguintes seções: na Seção 4.1 é efetuada uma contextualização da introdução de aspectos em sistemas em geral, na Seção 4.2 são apresentadas maneiras de automatizar a identificação de aspectos no desenvolvimento de sistemas, na Seção 4.3 são apresentados os conceitos de mineração de aspectos em sistemas orientados a objetos e as técnicas existentes, na Seção 4.4 são descritos os trabalhos que aplicam técnicas de agrupamento e que estão particularmente relacionados ao trabalho sendo proposto, na Seção 4.5 são apresentados os índices de avaliação de qualidade que foram propostos na literatura para a comparação dos algoritmos de agrupamento, na Seção 4.6 é apresentada uma compilação dos resultados de avaliações de algoritmos que foram encontrados na literatura e na Seção 4.7 são relacionadas as necessidades identificadas e oportunidades vislumbradas na mineração de aspectos que serviram de motivação para este trabalho.

#### 4.1 Introduzindo aspectos em sistemas

Como mencionado no Capítulo 2, todo sistema é composto de interesses base e interesses transversais. Os interesses base representam em sua essência os requisitos funcionais do software e são facilmente representados em modelos orientados a objetos. Em contrapartida, os interesses transversais representam os requisitos não funcionais, e portanto, a inclusão desses interesses em uma arquitetura de sistema orientado a objetos sem que haja perda das características de alta coesão e baixo acoplamento, é um desafio [37].

Considerando os impactos dos interesses transversais em software orientado a objetos, é facilmente identificável que a adoção da programação orientada a aspectos traz grandes benefícios ao ciclo de vida de um software. A utilização de aspectos faz com que os interesses base e os interesses transversais sejam claramente separados, contribuindo para que o sistema seja mais fácil de ser compreendido, e portanto, tornando as manutenções menos custosas e proporcionando facilidades na evolução do sistema quando comparadas a utilização única e exclusiva de orientação a objetos.

Para que um software possa usufruir dos benefícios da orientação a aspectos, é necessário que ele passe por um processo de decomposição de forma que os seus interesses transversais sejam separados de seus interesses base. Os passos típicos deste processo são [39]:

- Decomposição dos interesses (do inglês *Aspectual Decomposition*): Neste passo, as funcionalidades do software são decompostas para identificar quais fazem parte de interesses base e quais fazem parte de interesses transversais.
- Implementação dos interesses (do inglês *Concern Implementation*): Neste passo, interesses base e interesses transversais são implementados individualmente.

- **Recomposição dos interesses** (do inglês *Aspectual Recomposition*): Neste passo são especificadas as regras para a combinação através da criação de aspectos. O processo que efetua a recomposição, chamado de combinação, usa esta informação para compor o sistema final.

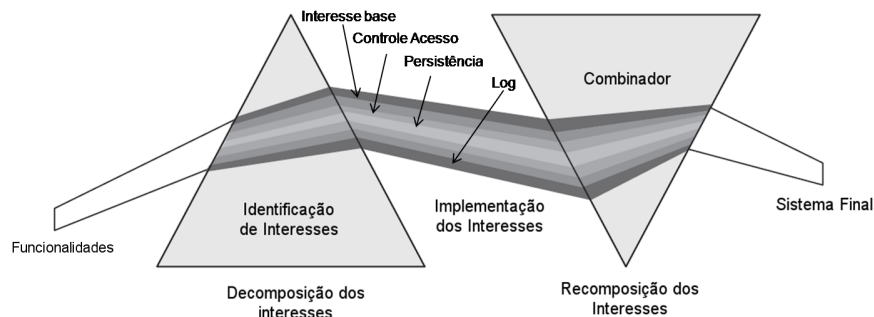


Figura 4.1: Processo de decomposição e integração de interesses de um software (adaptada de [39]).

A Figura 4.1 ilustra o processo em modo geral, tendo como entrada as funcionalidades do software, que é decomposto em interesses base e interesses transversais, que são implementados individualmente e ao final são integrados por um combinador para formar o sistema final.

## 4.2 Identificando aspectos

O processo de decomposição do software, quando feito de forma manual, exige grande esforço para a identificação e separação dos interesses, sendo uma tarefa árdua e trabalhosa. De forma a reduzir este esforço manual, surgiram três áreas de pesquisa com o objetivo de auxiliar neste processo [34]: (i) Mineração de Aspectos Antecipada; (ii) Navegadores Dedicados; e (iii) Mineração de Aspectos.

### Mineração de Aspectos Antecipada

A Mineração de Aspectos Antecipada (do inglês *Early Aspect Mining*) consiste na identificação dos interesses transversais candidatos a aspectos a partir de documentos de especificação do sistema. Esta técnica tenta identificar os candidatos a aspectos em um sistema antes mesmo da fase de construção, servindo como suporte aos analistas e arquitetos que podem identificar antecipadamente os aspectos, facilitando desta forma a modelagem e a arquitetura do novo software.

Uma técnica em avanço nesta área de pesquisa é o ACE (*Aspect Clustering Engine*, ou Motor de Agrupamento de Aspectos). O ACE, proposto em [16], utiliza um modelo probabilístico baseado na distribuição e recorrência de termos em fluxos dos casos de uso para computar as similaridades entre os diferentes requisitos e os agrupa através de um algoritmo de agrupamento hierárquico customizado.

Outra técnica existente, proposta por Sampaio et al [55], utiliza processamento de linguagem natural em diferentes fontes não estruturadas de documentos, como entrevistas ou documentos de requisitos, para descobrir palavras que são utilizadas em muitas sentenças. Os conjuntos de palavras que possuem alta frequência e têm o mesmo significado em todas as sentenças são considerados candidatos a aspectos.

## Navegadores Dedicados

Os Navegadores Dedicados (do inglês *Dedicated Browsers*) permitem a busca de interesses transversais candidatos a aspectos através da exploração assistida do código fonte do sistema legado. O ponto de partida do navegador é uma semente (do inglês *seed*), que é um elemento no código fonte que indica a presença de um determinado interesse transversal. A partir da semente, o desenvolvedor utiliza o navegador para explorar os módulos que estão relacionados a aquela semente, identificando desta maneira os interesses transversais existentes no código fonte. Alguns navegadores possuem recursos como linguagens específicas de consulta e repositórios de sementes [34].

Alguns exemplos de navegadores dedicados são: FEAT (*Feature Exploration and Analysis Tool*) [54]; ComSCId (*Computational Support for Concern Identification*) [50]; *Intentional View Browser* [45]; AMT (*Aspect Mining Tool*) [26]; Aspect Browser [22]; e Prism [71].

## Mineração de Aspectos

A Mineração de Aspectos (do inglês *Aspect Mining*) consiste na identificação dos interesses transversais candidatos a aspectos tipicamente a partir da mineração do código fonte de um sistema legado. A busca dos candidatos a aspectos é efetuada através da análise e identificação de determinados sintomas de interesses transversais. Várias técnicas foram propostas na literatura, como: análise de conceito formal, análise de agrupamento, análise de padrões, detecção de clones, entre outras que serão apresentadas na Seção 4.3.

Após a identificação dos interesses transversais candidatos a aspectos, é necessário efetuar a Refatoração para Aspectos (do inglês *Refactoring to Aspects*). A refatoração para aspectos visa a transformar os candidatos a aspectos identificados em aspectos reais [34].

A Figura 4.2 ilustra o processo completo de migração de um sistema orientado a objetos para orientado a aspectos. No primeiro momento são identificados os interesses transversais candidatos a aspectos no sistema legado, e, no segundo momento é efetuada a refatoração dos interesses transversais identificados para aspectos [34].

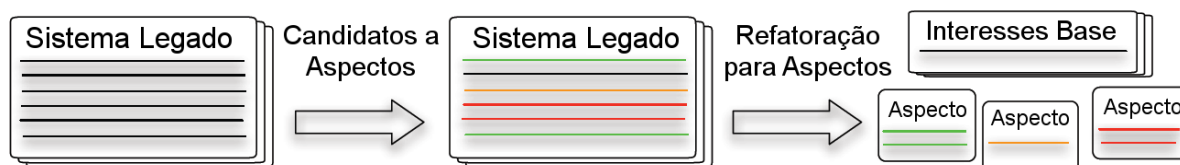


Figura 4.2: Migração de sistemas legados para aspectos (adaptada de [34]).

A área de pesquisa de mineração de aspectos é o tema principal deste trabalho e portanto é será aprofundada nas próximas seções.

## 4.3 Minerando aspectos

O processo de mineração de aspectos, de maneira geral, tenta encontrar os candidatos a aspectos através da análise estática do código fonte por sintomas de espalhamento ou

entrelaçamento de código, em um processo que têm por entrada o código fonte do sistema legado e como saída a identificação dos candidatos a aspectos.

As técnicas existentes não se limitam somente a análise estática do código fonte, existem por exemplo técnicas que efetuam análise estática de arquivos em repositórios de controle de versões, e também técnicas que efetuam mineração dinâmica em rastros de execução a partir do sistema em execução.

Quanto ao objeto de análise, existem técnicas que identificam candidatos a aspectos considerando fragmentos de métodos, métodos, classes ou arquivos de código fonte [34].

Um resumo das principais técnicas encontradas na literatura de Mineração de Aspectos é apresentado nas próximas subseções: (baseado em Kellens et al [34], com adaptações e atualizações).

### 4.3.1 Análise de *Fan-in*

Esta técnica surgiu das observações de Marin et al [43] que notaram que muitos dos interesses transversais exibem o comportamento de que seus métodos são invocados a partir de vários locais distintos, já que, suas funcionalidades são necessárias por diferentes métodos, classes e pacotes.

Em [43], é proposto o uso da métrica de *fan-in* de maneira a descobrir interesses transversais no código fonte. O *fan-in* de um método  $m$  é definido como a quantidade de corpos de métodos distintos que podem invocar  $m$ . No caso de polimorfismo, uma chamada para o método  $m$  contribui para aumentar o *fan-in* de todos os métodos derivados dos super tipos de  $m$ , bem como dos métodos das sub classes que refinam  $m$ . O algoritmo compreende os seguintes passos:

- Calcula a métrica de *fan-in* para todos os métodos do sistema.
- Filtra os resultados: Remove métodos de leitura e alteração de atributos de classes, exceto estáticos, bem como métodos utilitários, como por exemplo `toString()`. Também são removidos os métodos com um valor de *fan-in* abaixo de um determinado limite definido por parâmetro.
- Analisa manualmente os métodos remanescentes.

Em [72], Zhang et al. propôs a utilização da técnica de *Fan-In* combinada ao agrupamento de métodos baseado na similaridade de seus nomes.

### 4.3.2 Detecção de Clones

A técnica de detecção de clones parte do pressuposto de que um interesse transversal é implementado através do reúso de fragmentos de código [34]. A existência de duplicação de código pode ser um bom indício da existência de interesses transversais, já que estes não são claramente modularizados, e portanto, certas partes da implementação poderão exibir altos níveis de código duplicado para implementar estes interesses.

Três técnicas se apoiam nesta observação para efetuar a mineração de aspectos:

#### 1. Detecção de clones em grafos de dependências

Aplica detecção de clones em grafos de dependências de programa para detectar possíveis aspectos. Shepherd et al propuseram uma técnica em [60] que gera um grafo de dependências de programa em que os nós representam as instruções de

código e os vértices representam as relações de dependências de dados ou de controle entre as instruções correspondentes. De forma a identificar candidatos a aspectos, o algoritmo executa os seguintes passos:

- **Construção:** Constrói um grafo de dependências a partir do código fonte para todos os métodos.
- **Identificação:** Identifica um conjunto de candidatos a aspectos baseado nas estruturas de controle, comparando os métodos par a par.
- **Filtro:** Filtra o conjunto de candidatos a aspectos identificados como clones baseados na estrutura de controle, mantendo somente os que também possuem dependência de dados similares.
- **Junção:** Como segmentos de código duplicado podem ocorrer em mais de dois métodos, o que é comum em interesses transversais, junta os pares de métodos clones em conjuntos de candidatos similares.

De forma a reduzir a complexidade de comparação dos grafos, Shepherd et al citam que os adendos antes de métodos são os mais comuns de ocorrerem e, portanto, limitam a comparação ao início dos métodos. Os conjuntos finais de métodos são analisados manualmente para descobrir interesses transversais.

## 2. Detecção de clones baseada em símbolos

Esta técnica, proposta em [32], gera uma sequência de símbolos (*tokens*) a partir do código fonte através de um analisador léxico. Sobre as sequências geradas são aplicadas transformações com o propósito de transformar as porções de código em uma forma regular. A partir da forma regular o algoritmo compara as porções a fim de encontrar partes que possuem significado similar, independentemente da sintaxe.

Como resultado da comparação são obtidas as sequências que são consideradas clones uma da outra.

## 3. Detecção de clones baseada na árvore de sintaxe abstrata

Esta técnica identifica clones através da análise da árvore de sintaxe abstrata (do inglês *abstract syntax tree*) gerada a partir do código fonte.

Em [8], Baxter et al propõem uma heurística que primeiramente gera a árvore de sintaxe abstrata a partir de um analisador (*parser*) de código fonte e, por fim, aplica três algoritmos de identificação de clones: O primeiro detecta sub-árvores clones, o segundo detecta sequências de declarações e o terceiro tenta encontrar casos mais complexos de clones, como por exemplo, checar se os pais dos clones também são clones.

Como resultado são obtidos os grupos de fragmentos de código que são considerados clones um do outro.

### 4.3.3 Análise de repositórios de sistemas de controle de versões

Esta técnica foi proposta por Mulder [47] e consiste na extração de informações implícitas de repositórios de software gerenciados por sistemas de controle de versão. A técnica parte do princípio de que se determinados arquivos são frequentemente alterados juntos, este é um indício de que interesses transversais estão presentes no sistema.

Este princípio partiu da observação de que os desenvolvedores tendem a alterar várias entidades simultaneamente quando os elementos de um interesse estão espalhados pelo código do sistema. Esta técnica identifica acoplamento lógico, ou seja, dependências implícitas e evolucionárias entre artefatos de um sistema.

Para efetuar a mineração a técnica utiliza a extração de regras dos repositórios em dois níveis de granularidade: nível de arquivo ou nível de método. As regras que ocorrem com frequência são analisadas para a identificação dos candidatos a interesses transversais.

#### 4.3.4 Identificação baseada em catálogo de padrões

Esta técnica foi proposta por Figueiredo et al. [19] através da criação de uma ferramenta para busca de interesses transversais através da identificação de ocorrências de determinados padrões no código fonte.

A ferramenta utiliza um catálogo [18] que contém os padrões de ocorrências de interesses transversais que são frequentemente observados em sistemas reais. O catálogo é composto por 13 padrões, que são nomeados, descritos e classificados em 4 categorias. A ferramenta analisa o código fonte em busca destes padrões, identifica as ocorrências e classifica os interesses transversais conforme o catálogo.

#### 4.3.5 Classificação de métodos através de aprendizado de máquina

Esta técnica foi proposta por Shepherd et al [61] através da criação de um *framework* chamado Timna que utiliza classificação de métodos através de aprendizado de máquina. O aprendizado ocorre através da geração de regras de classificação a partir da combinação dos resultados obtidos pela aplicação de um conjunto de técnicas de Mineração de Aspectos e heurísticas específicas. No estudo [61] os autores utilizam nos experimentos os resultados das técnicas de *Fan-In* e Detecção de Clones, além de um conjunto de heurísticas composto por checagens de métodos sem parâmetros, métodos sem retorno, métodos com qualificadores *public*, *static* e *interface*, que possuem chamadas no início ou no final dos métodos que os invocam, entre outras.

Para que possam ser geradas as regras de classificação, o Timna executa um processo de treinamento composto por três fases:

1. Rótulo de Métodos: Nesta fase os métodos utilizados no treinamento são rotulados como candidatos e não candidatos a aspectos. Os candidatos a aspectos também são classificados em categorias: chamadas de métodos ordenadas, métodos de checagem, métodos de transformação, gatilhos/tratamento de eventos sem parâmetro, gatilho/tratamento de eventos com parâmetro, *Singletons*, *Factories*, adição ou remoção de *listeners*. Ao final desta fase é gerada a tabela de classificação.
2. Resultado da execução do conjunto de técnicas de Mineração de Aspectos: Nesta fase são coletados os resultados da aplicação de um conjunto de técnicas de Mineração de Aspectos. As análises geram resultados como booleanos, inteiros ou valores de porcentagem. Estes atributos são anexados a tabela de classificação formando a tabela de classificação estendida.
3. Aprendizado de Máquina: Nesta fase são aplicadas técnicas de aprendizado de máquina para a extração de regras a partir dos resultados das análises contidos na tabela de classificação estendida.

Após processo de treinamento, o Timna pode classificar métodos de qualquer outro sistema. O processo de classificação de métodos é composto de duas fases:

1. Resultado da execução do conjunto de técnicas de Mineração de Aspectos: Nesta fase é executado o conjunto de técnicas de mineração de aspectos no sistema alvo de análise. O conjunto de técnicas aplicado deverá ser o mesmo que foi utilizado no processo de treinamento. Os resultados das técnicas são coletados e relacionados na tabela de classificação estendida.
2. Classificação: Nesta fase é aplicado o classificador. O classificador recebe como entrada a tabela de classificação estendida e as regras extraídas na fase de treinamento e tem como saída a classificação dos métodos do sistema alvo.

### 4.3.6 Análise de padrões recorrentes em rastros de execução

Esta técnica analisa os rastros de execução (*execution traces*) de um programa em busca de padrões recorrentes de execução. Existem trabalhos propostos na literatura que aplicam esta técnica tanto dinamicamente (em tempo de execução) [9] [10] [65] [70] quanto estaticamente (a partir a análise do código fonte) [53].

Um exemplo de aplicação desta técnica é o *DynAMiT* [10] (*Dynamic Aspect Mining Tool*, ou Ferramenta de Mineração de Aspectos Dinâmica). O *DynAMiT* na primeira fase analisa os rastros de execução de um programa e efetua o mapeamento de todas as relações de invocações entre os métodos. Na segunda fase entra em ação o algoritmo de mineração que identifica os candidatos a aspectos verificando a existência de rastros de execução que ocorrem mais de uma vez, de forma uniforme e em diferentes contextos de chamada.

### 4.3.7 Análise de conceito formal

A análise de conceito formal utiliza um conjunto de objetos e atributos com as suas respectivas descrições como base para a geração de conceitos. As técnicas deste domínio utilizam heurísticas para identificar nestes conceitos gerados indicadores de interesses transversais. Há dois tipos de técnicas diferenciadas que utilizam análise de conceito formal:

#### 1. Análise de conceito formal de rastros de execução

Aplica análise formal nos rastros de execução de forma a descobrir possíveis aspectos. Um exemplo de aplicação desta técnica é o *Dynamo* [67].

O *Dynamo* utiliza na primeira fase uma versão instrumentada do sistema na qual são executados alguns casos de usos com o intuito de capturar os rastros de execução. Na segunda fase entra em ação o algoritmo de análise formal, sendo que os objetos são os próprios casos de uso e os atributos são os métodos que foram invocados durante a execução dos casos de uso. Como resultado, todos os conceitos que contêm rastros gerados exatamente por um caso de uso são selecionados. Estes serão considerados candidatos a aspectos se atenderem as seguintes restrições:

- Espalhamento: Os atributos (métodos) do conceito pertencem a mais de uma classe.

- Entrelaçamento: Diferentes métodos de uma mesma classe são especificados para mais de um caso de uso.

## 2. Análise de conceito formal de identificadores

Aplica análise formal em identificadores extraídos a partir do código fonte do sistema de forma a descobrir possíveis aspectos.

Em [11] é apresentada uma ferramenta que efetua a análise formal de identificadores que se baseia na crença de que os interesses transversais são implementados utilizando convenções de nomes nas classes e métodos do sistema. Nesta ferramenta, o algoritmo de conceito formal utiliza as classes e métodos como objetos, e, como atributos, o algoritmo utiliza cadeias de caracteres geradas a partir dos nomes dos métodos e das classes. Para a geração dessas cadeias de caracteres, o algoritmo separa os nomes significativos dos nomes dos métodos e das classes.

Por exemplo, uma classe chamada `ProdutoEstoque` é separada em sub cadeias de caracteres como `Produto` e `Estoque`. Sub cadeias pouco significativas, como `De` e `Com`, são descartadas dos resultados.

Os conceitos resultantes consistem no máximo de grupo de classes e métodos que compartilham o máximo número de partes da cadeia de caracteres. Apesar de o algoritmo filtrar muitos conceitos pouco significativos automaticamente, é possível que um número expressivo de conceitos continuem com a necessidade de inspeção manual.

Para identificação dos aspectos candidatos, os resultados são então restringidos a conceitos com indícios de interesses transversais, como por exemplo, métodos e classes pertencentes a pelos menos duas hierarquias de classes.

### 4.3.8 Processamento de linguagem natural no código fonte

Esta técnica utiliza o processamento de linguagem natural no código fonte de forma a tentar identificar candidatos a aspectos.

Shepherd et al propõem [63] uma abordagem que utiliza a informação de processamento de linguagem natural como um indicador de possíveis candidatos a aspectos baseada no pressuposto de que os interesses transversais são frequentemente implementados pelo rigoroso uso de convenções de nome e de códigos.

Eles utilizam no experimento a técnica de processamento de linguagem natural chamada de encadeamento léxico (do inglês *lexical chaining*) para encontrar grupos de entidades de código fonte relacionados que representam interesses transversais. O encadeamento léxico têm como entrada uma coleção de palavras, e como saída, as cadeias de palavras que são altamente relacionadas semanticamente.

De forma a criar as cadeias, o algoritmo requer uma medida de distância semântica entre cada combinação de palavras. Para este fim foi utilizada em [63] uma base de dados de relações conhecidas entre as palavras em combinação com a informação sobre a fala de cada palavra com o intuito de calcular o caminho semântico entre duas palavras.

De maneira a minerar por interesses transversais o algoritmo é aplicado em comentários, nomes de métodos, nomes de campos e nomes de classes do sistema. Ao final, também é necessária uma inspeção manual dos resultados para que se possa selecionar bons candidatos a aspectos.



### 4.3.9 Análise de Ligações

Huang et al [30] propõem uma técnica de mineração de aspectos baseada na análise de ligações (*link analysis*), técnica utilizada para ranqueamento de páginas da internet.

A técnica consiste na criação de um grafo de interesses que representa as ligações entre classes, utilizando para isto as dependências de herança e referência. A partir do grafo são computados o nível de espalhamento (*scatter*), que mede a quantidade de classes que referenciam determinada classe, e o nível de centralização (*centralization*), que mede a quantidade de classes que são referenciadas por uma determinada classe. Ao final, as classes são ranqueadas em ordem de tendência a serem interesses transversais.

O ranqueamento é baseado em três ideias:

- Implementação e Integração: Uma classe que possui um alto nível de espalhamento é considerada uma classe de implementação, em contraste, uma classe que possui um alto nível de centralização é considerada uma classe de integração. Classes de implementação possuem maior tendência a serem interesses transversais do que classes de integração.
- Frequência: Se uma classe é frequentemente utilizada pelo sistema, ela possui maior tendência a ser um interesse transversal do que uma classe menos utilizada.
- Quantidade de Saídas: Se determinadas ligações entre classes possuem uma grande quantidade de saídas (*Fan-Out*), a classe que possui as ligações de saída têm maior tendência a participar da lógica do núcleo do sistema do que ser uma candidata a aspecto.

### 4.3.10 Detecção de métodos únicos

Esta técnica surgiu de uma observação de Gybels et al [24] em que, antes da existência de orientação a aspectos, os interesses transversais eram implementados de maneira sistemática. Sendo assim, se este tipo específico de implementação for identificado, este poderá ser considerado um sintoma para um potencial candidato a aspecto.

Gybels et al citam que uma prática amplamente utilizada de implementação de interesses transversais é efetuada através da criação de uma entidade que é invocada a partir de diferentes módulos do sistema. De forma a detectar instâncias deste padrão, Gybels et al propuseram em [24] a heurística de Métodos Únicos (do inglês *Unique Methods*). Um método único é definido como:

“Um método sem um valor de retorno que implementa uma mensagem implementada por nenhum outro método”

Após calcular todos os métodos únicos do sistema, ordená-los conforme o número de vezes que o método é chamado, e filtrar métodos irrelevantes, como métodos de acesso (que somente fornecem operações de leitura ou alteração dos valores de atributos de classes), o usuário deve inspecionar manualmente todos os métodos resultantes de maneira a encontrar candidatos a aspectos apropriados.

Os autores demonstraram a aplicabilidade prática de sua técnica detectando alguns tipos de aspectos em uma imagem de *Smalltalk*, como a geração de rastros (*tracing*), atualização de notificações e gerenciamento de memória.

### 4.3.11 Baseada em grafos

Esta técnica tem como idéia básica a construção de um grafo para a identificação de interesses transversais.

Serban et al propõem uma abordagem em [58] na qual é gerado um grafo não direcionado a partir do código fonte em que os nós são os métodos do sistema e os vértices determinam a distância entre estes métodos. O algoritmo verifica a conectividade dos componentes do grafo de forma a gerar sub-grafos que representam partições de métodos relacionados.

Ao final as partições são ordenadas pela média da distância dos vértices e são analisadas manualmente como potenciais interesses transversais.

### 4.3.12 Análise de Agrupamento

As técnicas de análise de agrupamento aplicam algoritmos de agrupamento de forma a identificar grupos de métodos que possuem algum tipo de relacionamento entre si. Técnicas desse domínio baseiam-se no pressuposto de que métodos relacionados são indicadores da existência de interesses transversais.

Segundo [34] há dois tipos de técnicas que utilizam análise de agrupamento:

#### 1. Agrupamento baseado em métodos com nomes similares

Aplica análise de agrupamento de forma a agrupar os métodos que possuem nomes similares. Esta técnica identifica aspectos partindo do princípio de que métodos pertencentes a um interesse transversal possuem nomes similares.

Em [62], Shepherd et al reportam um experimento no qual utilizam agrupamento hierárquico aglomerativo para agrupar métodos relacionados. O experimento inicia colocando cada método em um grupo separado e, recursivamente, junta os grupos nos quais a distância entre os métodos é menor que um determinado limite definido por parâmetro.

No experimento inicial foi utilizada uma medida de distância que é inversamente proporcional ao tamanho da sub cadeia de caracteres comum dos nomes dos métodos, ou seja, quanto maior for a semelhança entre os nomes dos métodos, menor é a distância entre eles.

#### 2. Agrupamento baseado em invocações de métodos

Aplica análise de agrupamento de forma a agrupar os métodos que efetuam invocações de métodos em comum. Esta técnica identifica sintomas de espalhamento e entrelaçamento de código baseada no princípio de que, se os mesmos métodos são chamados frequentemente de diferentes módulos, esta é uma indicação de que um interesse transversal deve estar presente. Esta técnica é aplicada pelos trabalhos [12] [13] [27] [56] [57], os quais são apresentados com detalhes na próxima seção.

## 4.4 Aplicando análise de agrupamento na mineração de aspectos

Conforme apresentado na Seção 4.3, existem várias técnicas propostas para solucionar o problema de mineração de aspectos. Nesta seção é apresentada com detalhes a técnica de análise de agrupamento baseada em invocações de métodos, que é a base deste trabalho, incluindo a definição formal, o processo geral e as abordagens existentes na literatura.

### 4.4.1 Definição formal

A técnica de análise de agrupamento baseada em invocações de métodos segue de maneira geral o seguinte modelo formal, conforme proposto em [57]:

Um sistema de software  $S$  é definido como um conjunto de métodos  $\{m_1, m_2, \dots, m_n\}$ , onde  $m_i, 1 \leq i \leq n$ , é um método do sistema, e  $n$  representa o número de métodos de  $S$ .

Um interesse transversal é um conjunto de métodos  $IT_i = \{m_a, m_b, \dots, m_r\}$  que implementam um interesse, sendo  $r$  o número de métodos deste conjunto.

$IT = \{IT_1, IT_2, \dots, IT_q\}$  é o conjunto de todos os interesses transversais do sistema  $S$ , sendo  $q$  o número de interesses transversais de  $S$ .

Uma partição de  $S$  é formada pelo conjunto de partições  $K = \{K_1, K_2, \dots, K_p\}$ , onde cada partição deve conter ao menos um método, cada método deve pertencer a uma única partição, e da união de todas as partições deve-se obter o todo. Ou seja, uma partição é válida se  $1 \leq p \leq n$ ,  $K_i \subseteq S, K_i \neq \emptyset, \forall_i \in \{1, 2, \dots, p\}$ ,  $S = \bigcup_{i=1}^p K_i$  e  $K_i \cap K_j = \emptyset, \forall_i, j \in \{1, 2, \dots, p\}, i \neq j$ .

O problema de mineração de aspectos pode ser visto como um problema de encontrar um conjunto de partições  $K$  do sistema  $S$ . Uma partição de  $S$  é considerada ótima se todos os métodos pertencentes a um interesse transversal estão no mesmo grupo e os métodos pertencentes a este grupo são referentes ao mesmo interesse transversal.

### 4.4.2 Processo

O processo básico da mineração de aspectos através de análise de agrupamento baseada em invocação de métodos pode ser generalizado em 4 etapas, conforme proposto por Serban e Moldovan em [56]:

- Computação: Analisa o código fonte do sistema a fim de computar todas as classes, métodos e as relações de invocação entre eles.
- Filtro: Filtra métodos e invocações que devem ser ignorados.
- Agrupamento: Aplica um algoritmo de análise de agrupamento de maneira a formar grupos de métodos que possuem invocações em comum. Utiliza uma medida de similaridade que determina a distância entre dois métodos.
- Análise: Analisa os resultados obtidos de maneira a descobrir quais grupos contêm métodos pertencentes a um interesse transversal.

O processo apresentado é relacionado com o processo de agrupamento apresentado na Seção 3.1.1. Uma comparação das fases dos dois processos é apresentado na Tabela 4.1.

Tabela 4.1: Comparação do processo de Mineração de Aspectos e Agrupamento

	Mineração de Aspectos	Agrupamento
1	Computação	Seleção de características e extração
2	Filtro	Seleção de características e extração
3	Agrupamento	Seleção e projeto do algoritmo de agrupamento
4	Análise	Interpretação dos resultados

### 4.4.3 Abordagens existentes

As abordagens propostas na literatura utilizam principalmente a análise de agrupamento particional ou hierárquica de modo a formar grupos de métodos que possuem invocações em comum. De modo geral o problema é modelado da seguinte forma:

- Os métodos do sistema são os elementos que serão agrupados.
- A medida de distância entre os elementos considera em sua fórmula as invocações de métodos em comum entre os métodos avaliados.
- Os grupos formados representam conjuntos de métodos que possuem contextos de invocações similares.

São apresentadas nas próximas seções as seguintes abordagens classificadas por tipo de algoritmo de agrupamento utilizado:

- Particional
  - *k-means*: kAM [57].
  - *k-medoids*: PACO [12].
  - Algoritmos Genéticos: GAM [56].
- Hierárquico Aglomerativo
  - Clássico: HACO [13].
  - CHAMELEON: AMUCA [27].

#### 4.4.3.1 PACO

O PACO [12] (*Partitional Clustering Algorithm for Crosscutting Concerns Identification*, ou Algoritmo de Agrupamento Particional para Identificação de Interesses Transversais) é baseado no algoritmo de análise de agrupamento particional *k-medoids*.

#### Medidas de Distância

Foram utilizadas três medidas de distância com o algoritmo PACO:

- Distância de espalhamento: Esta função captura o sintoma de espalhamento de código. A ideia desta medida é que, se dois métodos são invocados por métodos ou classes comuns, eles devem pertencer ao mesmo grupo.

A distância  $D_S^{CCC}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é expressa na Equação 4.1:

$$D_S^{CCC}(m_i, m_j) = \begin{cases} 1 - \frac{|in(m_i) \cap in(m_j)|}{|in(m_i) \cup in(m_j)|}, & \text{se } in(m_i) \cap in(m_j) \neq \emptyset \\ \infty & \text{caso contrário} \end{cases} \quad (4.1)$$

Onde  $in(m)$  é uma coleção que consiste de: o método  $m$ , a classe na qual o método  $m$  é definido e as classes e métodos que invocam  $m$ . Caso existam classes aninhadas (classes declaradas dentro de uma classe) será contabilizada também a classe principal que contém a classe aninhada.

- Distância de entrelaçamento: Esta função captura o sintoma de entrelaçamento de código. A ideia desta medida é que, se dois métodos são invocados em contextos similares, eles devem pertencer ao mesmo grupo.

A distância  $D_T^{CCC}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é expressa na Equação 4.2:

$$D_T^{CCC}(m_i, m_j) = \begin{cases} 1 - \frac{|r(m_i) \cap r(m_j)|}{|r(m_i) \cup r(m_j)|}, & \text{se } r(m_i) \cap r(m_j) \neq 0 \\ \infty & \text{caso contrário} \end{cases} \quad (4.2)$$

Onde  $r(m)$  é uma coleção que consiste de: variáveis locais que possuem declaração e uso, atributos utilizados, classes acessadas e tipos de parâmetros utilizados pelo método invocador.

A ideia desta medida é coletar toda a informação que pode ser considerada como o contexto de invocação do método  $m$ .

- Distância de espalhamento-entrelaçamento: Esta função tenta capturar tanto o sintoma de espalhamento de código quanto o de entrelaçamento de código. A ideia principal é unir a função de espalhamento com a de entrelaçamento. A distância  $D_{ST}^{CCC}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é expressa na Equação 4.3:

$$D_{ST}^{CCC}(m_i, m_j) = \min \{ D_S^{CCC}(m_i, m_j), D_T^{CCC}(m_i, m_j) \} \quad (4.3)$$

Onde  $D_{ST}^{CCC}(m_i, m_j)$  recebe o valor mínimo obtido pela aplicação das medidas  $D_S^{CCC}(m_i, m_j)$  e  $D_T^{CCC}(m_i, m_j)$ .

### Descrição do algoritmo

O PACO utiliza o algoritmo *k-medoids* tradicional, porém, com uma heurística própria para a escolha dos *medoids* iniciais e para definir a quantidade de *medoids*. A heurística utilizada pelo PACO é a seguinte:

1. A quantidade inicial  $p$  de grupos é  $n$  (a quantidade de métodos do sistema).
2. O método escolhido como o primeiro *medoid* é o método mais distante do conjunto de todos os métodos, ou seja, o método que maximiza a soma das distâncias de todos os outros métodos.
3. Para cada um dos métodos remanescentes (que não foram escolhidos como *medoids*), é computada a distância entre o método e o *medoid* mais próximo. O próximo *medoid* que será escolhido é o método  $m$  que maximiza a distância entre o método e seu *medoid* mais próximo e que sua distância seja maior que o limite estipulado pelo parâmetro *distMin*. Se não existir método com distância maior que *distMin*, significa que  $m$  está muito perto de seu *medoid* e, portanto, não deve ser escolhido como um novo *medoid*. Neste caso o número  $p$  de grupos é decrementado.
4. O Passo 3 é executado repetidamente até que o número de *medoids* escolhidos seja igual a quantidade de grupos  $p$ .

O algoritmo utiliza os seguintes passos para agrupar os métodos de um sistema:

1. O número inicial  $p$  de grupos e os *medoids* iniciais são determinados pela heurística citada anteriormente.

2. Os grupos são recalculados. Cada objeto é atribuído ao *medoid* mais próximo.
3. Recalcula o *medoid*  $i$  de cada grupo  $k$  baseado na seguinte ideia: Se  $h$  é um objeto de  $k$  tal que a saída da Função 4.4 seja negativa, então  $h$  se torna o novo *medoid* do grupo  $k$ .

$$\sum_{j \in k} (d(j, h) - d(j, i)) \quad (4.4)$$

4. Os Passos 2 e 3 são repetidamente executados até que não haja mais alterações na partição  $K$ .

Se em algum momento um grupo se tornar vazio, o número de grupos será decrescido.

### 4.4.3.2 *kAM*

O *kAM* [57] (*k-means in Aspect Mining*, ou *k-means* em Mineração de Aspectos) é baseado no algoritmo de análise de agrupamento particional *k-means*.

#### Medidas de Distância

A medida de distância utilizada pelo *kAM* é a tradicional Distância Euclidiana. A distância  $D_E^{SM}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é expressa na Equação 4.5.

$$D_E^{SM}(m_i, m_j) = \sqrt{\sum_{k=1}^l (m_{ik} - m_{jk})^2} \quad (4.5)$$

Cada método  $m$  do sistema é representado por um vetor  $l$ -dimensional onde:  $m_i = (m_{i1}, \dots, m_{il})$ ,  $l$  é o total de dimensões,  $1 \leq k \leq l$  e  $i \neq j$ .

Serban et al utilizaram dois tipos distintos de modelos de espaço vetorial com o *kAM* [57]:

- O vetor associado com o método  $m$  é  $\{FIV, CC\}$ , onde *FIV* é o *fan-in* de  $m$  e *CC* é o número de classes que invocam  $m$ .
- O vetor associado com o método  $m$  é  $\{FIV, C_1, C_2, \dots, C_l\}$ , onde *FIV* é o *fan-in* de  $m$ ,  $l$  é o total de classes existentes no sistema, e  $C_i$  ( $1 \leq i \leq l$ ) é 1, se o método  $m$  é invocado por um método pertencente a classe  $C_i$ , e 0 caso contrário.

#### Descrição do algoritmo

O *kAM* utiliza o algoritmo *k-means* clássico, porém, com uma heurística para a escolha do número de grupos e centróides iniciais, conforme apresentada abaixo:

1. O número inicial  $k$  de grupos é  $n$  (o número de métodos do sistema).
2. O método escolhido como o primeiro centróide é o método mais distante do conjunto de todos os métodos (o método que maximiza a soma das distâncias de todos os outros métodos).

3. Para cada um dos métodos remanescentes (que não foram escolhidos como centróides), é computada a distância entre o método e o centróide mais próximo. O próximo centróide que será escolhido é o método  $m$  que maximiza a distância entre o método e seu centróide mais próximo e que sua distância seja maior que o limite estipulado pelo parâmetro  $distMin$ . Se não existir método com distância maior que  $distMin$ , significa que  $m$  está muito perto de seu centróide e, portanto, não deve ser escolhido como um novo centróide. Neste caso o número  $k$  de grupos será decrescido.
4. O Passo 3 é executado repetidamente até que o número de centróides escolhidos seja igual a quantidade de grupos  $k$ .

Após a execução da heurística para a escolha dos centróides iniciais, o  $kAM$  se comporta exatamente como o algoritmo  $k-means$  clássico. O algoritmo termina quando os grupos permanecem sem alteração durante duas iterações ou quando o número de passos executados excede o número máximo de iterações permitidos.

### 4.4.3.3 HACO

O HACO [13] (*Hierarchical Clustering Algorithm for Crosscutting Concerns Identification*, ou Algoritmo de Agrupamento Hierárquico para Identificação de Interesses Transversais) é baseado no algoritmo de análise de agrupamento hierárquico aglomerativo.

#### Medida de distância

A medida de distância  $D_S^{CCC}(m_i, m_j)$  utilizada pelo HACO para computar a distância entre dois métodos  $m_i$  e  $m_j$  é definida na Equação 4.6:

$$D_S^{CCC}(m_i, m_j) = \begin{cases} 0 & i = j \\ 1 - \frac{|Col(m_i) \cap Col(m_j)|}{|Col(m_i)| + |Col(m_j)|} & \text{se } Col(m_i) \cap Col(m_j) \neq \emptyset \\ \infty & \text{caso contrário} \end{cases} \quad (4.6)$$

Onde  $Col(m)$  é uma coleção que consiste de: o método em si, a classe na qual o método é definido, as classes e os métodos que invocam  $m$  e as classes nas quais as classes e métodos que invocam  $m$  estão contidos.

#### Descrição do algoritmo

O HACO utiliza uma heurística que determina o número de grupos que serão formados. A heurística tem por finalidade identificar  $p$  métodos representativos do sistema  $S$  que representarão os grupos. A heurística possui o seguinte funcionamento:

- O método escolhido como o primeiro método representativo é o método mais “distante” do conjunto de todos os métodos (o método que maximiza a soma das distâncias de todos os outros métodos).
- A cada passo é selecionado, a partir dos métodos remanescentes, o método mais distante relativo aos métodos já escolhidos. Se o método selecionado possui distância menor que o parâmetro  $distMin$  de um método representativo já escolhido, o processo termina, caso contrário, o método selecionado é considerado um novo método representativo e o processo é repetido novamente.

Após a identificação do número de grupos, o HACO aplica o algoritmo de agrupamento hierárquico aglomerativo clássico. É utilizada a medida de ligação *complete linkage*, em que a distância entre dois grupos é determinada pelos dois elementos mais distantes.

Os passos principais executados pelo HACO são:

1. Cada método do sistema é incluído em seu próprio grupo.
2. Os seguintes passos são repetidos até que  $k$  grupos sejam atingidos ( $k$  é o número de grupos identificados pela heurística).
  - (a) Seleciona os dois grupos  $K_i$  e  $K_j$  mais similares a partir da partição atual, ou seja, o par de grupos que minimiza a distância entre eles.
  - (b) Os grupos  $K_i$  e  $K_j$  serão mesclados, ou senão, a partição continua inalterada.

#### 4.4.3.4 GAM

O GAM [56] (*Genetic Aspect Mining*, ou Mineração de Aspectos Genética) utiliza análise de agrupamento baseado em algoritmos genéticos em sua heurística.

##### Medida de Distância

O GAM, assim como o  $kAM$ , utiliza a distância Euclidiana como medida de distância. A distância  $D_E^{SM}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é expressa na Equação 4.7.

$$D_E^{SM}(m_i, m_j) = \sqrt{\sum_{k=1}^l (m_{ik} - m_{jk})^2} \quad (4.7)$$

Cada método  $m$  do sistema é representado por um vetor  $l$ -dimensional onde:  $m_i = (m_{i1}, \dots, m_{il})$ ,  $l$  é o total de dimensões,  $1 \leq k \leq l$  e  $i \neq j$ .

O vetor associado ao método  $m$  é  $\{FIV, C_1, C_2, \dots, C_l\}$ , onde  $FIV$  é o *fan-in* de  $m$ ,  $l$  é o total de classes existentes no sistema, e  $C_i$  ( $1 \leq i \leq l$ ) é 1, se o método  $m$  é invocado por um método pertencente a classe  $C_i$ , e 0 caso contrário.

##### Função de Avaliação

O GAM utiliza o algoritmo genético clássico e utiliza como função de avaliação a minimização do erro quadrático médio da dispersão do agrupamento conforme definido na Equação 4.8:

$$SSE(\mathcal{K}) = \sum_{j=1}^p \sum_{i=1}^{n_j} d^2(m_i^j, f_j) \quad (4.8)$$

Onde  $K = \{K_1, K_2, \dots, K_p\}$  é uma partição do sistema  $S$ , o grupo  $K_j$  é um conjunto de métodos  $\{m_1^j, m_2^j, \dots, m_{n_j}^j\}$  e  $f_j$  é o centróide de  $K_j$ , calculado conforme a Equação 4.9.

$$f_j = \left( \frac{\sum_{k=1}^{n_j} m_{k1}^j}{n_j}, \dots, \frac{\sum_{k=1}^{n_j} m_{kl}^j}{n_j} \right) \quad (4.9)$$



A medida  $SSE$  é utilizada como função de avaliação do GAM, significando que, se  $K$  é uma partição gerada por um indivíduo  $i$ , a função de avaliação de  $i$  é:

$$avaliacao(i) = Max - SSE(K) \quad (4.10)$$

Onde,  $Max$  é o valor máximo do erro quadrático médio.

### Descrição do Algoritmo

O GAM utiliza a seguinte heurística para escolher o número de grupos:

1. O número inicial  $p$  de grupos é  $n$  (o número de métodos do sistema).
2. O método escolhido como o primeiro centróide é o método mais distante do conjunto de todos os métodos (o método que maximiza a soma da distância de todos os outros métodos).
3. Para cada um dos métodos remanescentes (que não foram escolhidos como centróides), é computada a distância entre o método e o centróide mais próximo. O próximo centróide que será escolhido é o método  $m$  que maximiza a distância entre o método e seu centróide mais próximo e cuja distância seja maior que o limite estipulado pelo parâmetro  $distMin$ . Se não existir método com distância maior que  $distMin$ , significa que  $m$  está muito perto de seu centróide e, portanto, não deve ser escolhido como um novo centróide. Neste caso o número  $k$  de grupos será decrescido.
4. O Passo 3 é executado repetidamente até que o número de centróides escolhidos seja igual a quantidade de grupos  $k$ .

Dada a heurística acima, a população inicial é gerada, sendo que cada indivíduo  $i$  é constituído de um vetor de inteiros de tamanho fixo (que é igual ao número de métodos do sistema,  $n$ ). Cada posição define a adesão do método aos grupos. Portanto  $i = (i_1, i_2, \dots, i_n)$ ,  $1 \leq i_j \leq p$ ,  $\forall j, 1 \leq j \leq n$ , onde  $p$  é o número de grupos obtidos pela heurística de definição do número de grupos e  $i_j = r$  se o método  $m_j$  pertence ao grupo  $K_r$ .

A ideia principal do GAM é a execução dos seguintes passos até que o número máximo de gerações seja atingido:

1. Inclui na próxima geração os melhores indivíduos  $b$  da geração atual. Sendo  $b$  um parâmetro de entrada.
2. Dois pais são selecionados utilizando o método de seleção de roleta até que seja obtida uma geração completamente nova.
3. Os pais são recombinados, utilizando a recombinação em único ponto.
4. Os filhos sofrem mutação, substituindo um gene selecionado aleatoriamente por um valor aleatório entre 1 e  $p$ .
5. Os filhos são incluídos na próxima geração.

Após o número máximo de gerações ser atingido, o melhor indivíduo obtido da população é escolhido. Este indivíduo irá determinar a partição ótima do sistema  $S$ .

### 4.4.3.5 AMUCA

O AMUCA [27] possui uma heurística que utiliza tanto agrupamento quanto extração de regras, sendo que a fase de análise de agrupamento é efetuada através do algoritmo hierárquico aglomerativo CHAMELEON [33].

#### Medida de Distância

Cada método  $m$  do sistema é representado por um vetor  $l$ -dimensional onde:  $m_i = (m_{i1}, \dots, m_{il})$ ,  $l$  é o número de métodos do sistema. Para cada método  $m_i$  e  $m_j$ , se o método  $m_j (k = 1 \dots l)$  invoca um método  $m_i$ , então o valor da dimensão  $k$  do objeto  $m_i$  é 1, senão é 0.

A distância  $D^{HBZH}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é expressada na Equação 4.11:

$$D^{HBZH}(m_i, m_j) = \frac{r + s}{q + r + s} \quad (4.11)$$

Onde  $q$  o número de dimensões cujos valores de ambos  $m_i$  e  $m_j$  são 1.  $r$  o número de dimensões cujo valor do método  $m_j$  é 1 e o valor do método  $m_i$  é 0, e  $s$  o número de dimensões cujo valor do método  $m_i$  é 1 e o valor do método  $m_j$  é 0.

#### Descrição do Algoritmo

O AMUCA efetua a mineração de aspectos em 6 fases:

1. Construção da matriz de dados e dissimilaridades: Gera a matriz de dados e dissimilaridades a partir dos métodos do sistema, conforme descrito na medida de distância.
2. Agrupamento: Aplica o algoritmo de agrupamento hierárquico aglomerativo CHAMELEON [33].
3. Filtro de candidatos a aspectos: Os grupos resultantes após a aplicação do CHAMELEON são filtrados, mantendo-se somente os grupos cujos métodos possuam frequência de invocação superior a um dado limite.
4. Construção do conjunto de itens e do conjunto de transações: Cria um conjunto de itens e um conjunto de transações com o intuito de gerar um mapeamento para identificar os pontos de junção entre os interesses transversais e os interesses base.

O conjunto de itens  $I = (i_1, i_2, \dots, i_l, i_{l+1})$  é constituído pelos  $m$  métodos do sistema e é estendido com duas funções,  $\text{inicio}()$  e  $\text{fim}()$ , que foram introduzidas para localização.

Uma transação é formada por todos os itens invocados por um determinado método do sistema, recebe um TID, que é um identificador de transação, e por padrão chama o item  $\text{inicio}()$  antes da execução do método e o item  $\text{fim}()$  após a execução do método. O conjunto de dados é determinado por  $D = T_1, T_2, \dots, T_l$  e inclui todas as transações do sistema, sendo que para cada  $T_j, j = (1, 2, \dots, l), T_j \subseteq I^*$ .

5. Obtenção das regras de associação: As regras de associação são obtidas após a execução do algoritmo de mineração de regras Apriori [1]. São utilizadas como entradas o conjunto de transação, o conjunto de itens, os valores do grau de confiança e o grau de suporte. Durante o processo de mineração de regras de

associação são computados os conjuntos de itens que ocorrem com frequência, sendo que o objetivo é encontrar o relacionamento cujo método B é chamado logo após a chamada do método A, isto é, SE A ENTÃO B.

6. Filtragem para produzir o conjunto de regras final: Após a obtenção das regras pelo algoritmo Apriori, uma série de regras similares com SE A ENTÃO B são produzidas. Estas regras somente provam que A e B surgem ao mesmo tempo, porém não confirma se o método B é chamado logo após A ou não, o que é considerado um ponto chave de um adendo. Portanto, atributos adjuntos entre itens de transação são utilizados para filtrar as regras inválidas. Sendo assim, a regra SE A ENTÃO B somente será válida se o método B surge logo após o método A.

O AMUCA tem como saída o conjunto dos grupos de candidatos a aspectos e o conjunto de regras no formato SE A ENTÃO B.

Na próxima seção são apresentados alguns índices existentes na literatura que proporcionam a avaliação e comparação da qualidade dos resultados obtidos pelos algoritmos de mineração de aspectos baseados em análise de agrupamento.

## 4.5 Avaliando a qualidade da mineração de aspectos

Para a avaliação dos algoritmos de mineração de aspectos baseados em análise de agrupamento, foram propostos na literatura alguns índices de qualidade com o intuito de criar modos de comparação dos resultados dos algoritmos, dos quais podem-se citar:

- **ACC** (*Accuracy of a clustering based aspect mining technique*) ou acurácia de uma técnica de mineração de aspectos baseada em agrupamento: é um índice de qualidade proposto em [56] definido da seguinte maneira:

Sendo  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista e  $\mathcal{K}$  o conjunto de candidatos a aspectos encontrados pelo algoritmo de agrupamento, o índice de qualidade  $ACC(\mathcal{K}, IT)$  de um conjunto de grupos de candidatos a aspectos  $\mathcal{K}$  em respeito ao conjunto de instâncias de interesses transversais  $IT$  é definido na Equação 4.12:

$$ACC(\mathcal{K}, IT) = \frac{1}{q} \sum_{i=1}^q acc(\mathcal{K}, IT_i). \quad (4.12)$$

Sendo  $q$  a quantidade de instâncias de interesses transversais de  $IT$ ,  $IT_i$  uma instância específica de interesse transversal pertencente ao conjunto  $IT$  e  $acc(\mathcal{K}, IT_i)$  a precisão da instância de interesse transversal  $IT_i$ , definida como:

$$acc(\mathcal{K}, IT_i) = \begin{cases} \frac{\|IT_i \cap \mathcal{K}_j\|}{\|IT_i\|} & , \text{ se } \mathcal{K}_j \text{ é o } 1^\circ \text{ grupo na qual } IT_i \text{ foi descoberto} \\ 0 & , \text{ caso contrário} \end{cases} \quad (4.13)$$

A saída deste índice é a proporção de métodos de  $IT_i$  que surgem no primeiro grupo onde  $IT_i$  foi descoberto e é compreendido no intervalo entre 0 e 1. Quanto maior o valor de  $ACC$ , melhores são as partições em respeito a  $IT$ , indicando que o índice  $ACC$  deve ser maximizado.

- **PAME** (*Percentage of Analyzed Methods for a partition*), ou porcentagem de métodos analisados de uma partição: é um índice de avaliação de qualidade proposto em [56] definido da seguinte maneira:

Sendo  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista e  $\mathcal{K}$  o conjunto de candidatos a aspectos encontrados pelo algoritmo de agrupamento, o índice de qualidade  $PAME(IT, \mathcal{K})$  de um conjunto de grupos de candidatos a aspectos  $\mathcal{K}$  em respeito ao conjunto de instâncias de interesses transversais  $IT$  é definido na Equação 4.14:

$$PAME(IT, \mathcal{K}) = \frac{1}{q} \sum_{i=1}^q pame(IT_i, \mathcal{K}) \quad (4.14)$$

Sendo  $q$  a quantidade de instâncias de interesses transversais de  $IT$ ,  $IT_i$  uma instância de interesse transversal específica pertencente ao conjunto  $IT$  e  $pame(IT_i, \mathcal{K})$  a porcentagem mínima de métodos que precisam ser analisados do conjunto de candidatos  $\mathcal{K}$  para identificar uma instância de interesse transversal  $IT_i$ , definida como:

$$pame(IT_i, \mathcal{K}) = \frac{1}{|M|} \sum_{j=1}^r |\mathcal{K}_j| \quad (4.15)$$

onde  $r = \max \{t | 1 \leq t \leq p \text{ e } \mathcal{K}_t \cap IT_i \neq \emptyset\}$  é o índice do ultimo grupo do conjunto  $\mathcal{K}$  que contém métodos de  $IT_i$ .

A saída deste índice é a porcentagem do número de métodos que precisam ser analisados no conjunto de candidatos a aspectos de maneira a identificar todos os interesses transversais que estão no sistema  $S$  e é compreendida no intervalo entre 0 e 1. Quanto menor o valor de  $PAME$ , indica um menor tempo para análise, significando que  $PAME$  deve ser minimizado.

- **DISP** (*DISPersion of crosscutting concerns*) ou Dispersão de interesses transversais: Este índice é descrito em [21] e define o grau de dispersão de interesses transversais em grupos, considerando, para cada instância de interesse transversal, o número de grupos que contém elementos pertencentes a um interesse.

Sendo  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista e  $\mathcal{K}$  o conjunto de grupos de candidatos a aspectos encontrados pelo algoritmo de agrupamento. O índice de qualidade  $DISP(IT, \mathcal{K})$  de um conjunto de instâncias de interesses transversais  $IT$  em respeito a um conjunto de candidatos a aspectos  $\mathcal{K}$  é definido na Equação 4.16:

$$DISP(IT, \mathcal{K}) = \frac{1}{q} \sum_{i=1}^q disp(IT_i, \mathcal{K}) \quad (4.16)$$

Onde  $q$  é quantidade de interesses transversais de  $IT$ ,  $IT_i$  é um interesse pertencente ao conjunto  $IT$  e  $disp(IT_i, \mathcal{K})$  é a dispersão de uma instância de interesse transversal  $IT_i$ , definida como:

$$DISP(IT_i, \mathcal{K}) = \frac{1}{|D_{IT_i}|} \quad (4.17)$$

Onde  $D_{IT_i}$  é o conjunto de grupos que contêm elementos pertencentes a instância de interesse transversal  $IT_i$ , sendo que:

$$D_{IT_i} = \{k | k \in \mathcal{K} \text{ e } k \cap IT_i \neq \emptyset\} \quad (4.18)$$

A saída deste índice é a média da dispersão das instâncias dos interesses transversais em grupos e é compreendida no intervalo entre 0 e 1. Quanto maior o valor, mais coesas são as partições obtidas pelo algoritmo de agrupamento, indicando que DISP deve ser maximizado.

- **DIV** (*DIVersity of a partition*) ou Diversidade de uma partição: Este índice é descrito em [21] e define o grau no qual cada grupo contêm elementos de interesses transversais diferentes.

Sendo  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista e  $\mathcal{K}$  o conjunto de candidatos a aspectos identificados pelo algoritmo. A diversidade de um conjunto  $\mathcal{K}$  em respeito ao conjunto  $IT$ , denotada por  $DIV(IT, \mathcal{K})$ , é definida como:

$$DIV(IT, \mathcal{K}) = \frac{1}{p} \sum_{i=1}^p div(IT, \mathcal{K}_i) \quad (4.19)$$

Onde  $p$  é quantidade de grupos de candidatos a aspectos do conjunto  $\mathcal{K}$ ,  $\mathcal{K}_i$  é um grupo candidato a aspecto pertencente ao conjunto  $\mathcal{K}$  e  $div(IT, \mathcal{K}_i)$  é a diversidade de um grupo  $\mathcal{K}_i \in \mathcal{K}$ , definido como:

$$div(IT, \mathcal{K}_i) = \frac{1}{|V_{\mathcal{K}_i}| + \tau(\mathcal{K}_i)} \quad (4.20)$$

Onde,

$$V_{\mathcal{K}_i} = \{IT_j | IT_j \in IT \text{ e } \mathcal{K}_i \cap IT_j \neq \emptyset\} \quad (4.21)$$

é o conjunto de interesses transversais que possuem elementos em  $\mathcal{K}_i$ , e

$$\tau(\mathcal{K}_i) = \begin{cases} 1 & \text{se } \mathcal{K}_i \cap IT \neq \emptyset \\ 0 & \text{se } \mathcal{K}_i \cap IT = \emptyset \end{cases} \quad (4.22)$$

$\tau(\mathcal{K}_i)$  é 1 se o grupo  $\mathcal{K}_i$  contém elementos que não implementam interesses transversais e 0 caso contrário.

A saída deste índice é a média da diversidade dos grupos encontrados pelo algoritmo de agrupamento e é compreendida no intervalo entre 0 e 1. Quanto mais alto o valor de DIV, melhores são as partições em respeito ao conjunto de instâncias de interesses transversais que foram identificadas, indicando que DIV deve ser maximizado.

Na próxima seção serão apresentadas as avaliações de algoritmos existentes na literatura com os índices apresentados nesta seção.

## 4.6 Resultados encontrados na literatura

Nesta seção são apresentados os resultados que foram encontrados na literatura de mineração de aspectos.

A Tabela 4.2 compila os resultados dos algoritmos de mineração de aspectos baseados em análise de agrupamento listados na Seção 4.4 que possuem avaliação com ao menos um dos índices de qualidade listados na Seção 4.5. Para o caso do AMUCA, não foi localizado na literatura nenhum trabalho que avalie a qualidade dos grupos obtidos por este algoritmo.

Tabela 4.2: Comparação dos algoritmos existentes na literatura

Algoritmo	Sistema	Medida	DISP	DIV	ACC	PAME	Referência
<i>kAM</i>	JHotDraw	$D_E^{SM}$	0.4005	0.9972	—	—	[21]
	Laffra	$D_E^{SM}$	—	—	0.667	0.200	[56]
PACO	JHotDraw	$D_S^{CCC}$	0.4444	0.9753	—	—	[21]
	JHotDraw	$D_T^{CCC}$	0.4433	0.8732	—	—	[21]
	JHotDraw	$D_{ST}^{CCC}$	0.4207	0.8798	—	—	[21]
HACO	JHotDraw	$D_S^{CCC}$	0.4570	1.0000	—	—	[21]
GAM	Laffra	$D_E^{SM}$	—	—	0.667	0.220	[56]
AMUCA	—	$D^{HBHZ}$	Não há avaliação publicada				

Conforme pode ser observado na Tabela 4.2, o algoritmo HACO obteve os melhores resultados na avaliação do sistema JHotDraw quando na aplicação dos índices DISP e DIV. Em [21] os autores afirmam o desempenho superior apresentado pelo HACO e informam que das técnicas de agrupamento avaliadas, o agrupamento hierárquico é aparentemente mais adequado para a mineração de aspectos do que o agrupamento particional baseado nos critérios de medição criados pelos próprios autores.

Na próxima seção são apresentadas algumas limitações identificadas nos trabalhos existentes na literatura sobre a mineração de aspectos baseada em agrupamento.

## 4.7 Limitações identificadas

Analisando os trabalhos relacionados, foram identificadas algumas limitações que não foram contempladas. A partir destas limitações foram elencados alguns itens que são explorados no presente trabalho afim de contribuir para a área de mineração de aspectos baseada em agrupamento.

1. Metodologia de avaliação padronizada: A existência de padrões distintos de avaliação prejudica a comparação direta entre os algoritmos de agrupamento, medidas de distância e sistemas avaliados. Portanto é necessária a adoção de uma metodologia padronizada, e, que esta seja aplicada na comparação de todas as possibilidades de combinações de algoritmos, medidas e sistemas.
2. Avaliações realizadas com poucos sistemas: Um bom resultado obtido em um sistema não garante que o bom resultado seja obtido em outros sistemas. Portanto

é necessário que os algoritmos sejam submetidos para avaliação com vários sistemas que possuam propósitos distintos e que tenham sido construídos por diferentes desenvolvedores.

3. Aperfeiçoamento das medidas de distância existentes: A medida de distância é um dos itens que mais influencia os resultados dos algoritmos de agrupamento. Uma boa medida de distância deve ser capaz de capturar as características relevantes dos elementos que serão agrupados. Sendo assim, é considerada a oportunidade de aprimorar as medidas de distância existentes e explorar outros sintomas relevantes através do aproveitamento de ideias de outras técnicas existentes, como a detecção de métodos clones e agrupamento de métodos com nomes similares.
4. Uso do *fan-in* na fase de filtro: A técnica de *fan-in* possui um modelo que permite identificar o impacto que cada método causa no espalhamento do sistema. Métodos com baixo *fan-in*, por terem característica de baixo espalhamento, podem ser eliminados antes mesmo da fase de agrupamento, reduzindo o universo de métodos que precisam ser agrupados e conseqüentemente o tempo de processamento dos algoritmos de agrupamento.
5. Ordenação dos grupos: O processo geral existente na literatura de mineração de aspectos baseada em agrupamento não prevê uma maneira de ordenar os grupos conforme o potencial de serem candidatos a aspectos. A ausência de tal ordenação faz com que o analista receba uma centena de grupos, dos quais, terá que analisar um a um sem a possibilidade de definir um critério de prioridade de análise. Este fato impacta em baixa produtividade, já que o analista perde muito tempo com a análise de grupos que tem baixa probabilidade de serem aspectos. Portanto, é vislumbrado a incorporação de uma fase de ordenação de grupos ao processo geral e a definição de medidas de ordenação que permitam o cálculo de uma pontuação individual dos grupos de forma a ordená-los por ordem de potencialidade de serem aspectos.
6. Identificação de pontos de corte: Um aspecto é basicamente composto por adendos e a definição de pontos de corte que entrecortam certos pontos de junção. Um candidato a aspecto indica quais operações deverão ser refatoradas para um adendo, porém, não indica quais pontos de corte deverão existir para a junção dos interesses transversais com os interesses base. É vislumbrada a criação de um método automatizado para a identificação dos pontos de corte, que indique quais são os possíveis candidatos a pontos de corte para os candidatos a aspectos identificados.
7. *Framework* de implementação: De forma a transformar a teoria em prática, deve ser definido um *framework* de implementação de todo o processo para uma linguagem de programação que suporte orientação a aspectos. O objetivo do *framework* é se tornar um modelo de referência de implementação onde conste todos os detalhes técnicos necessários para a implementação do processo, e que o mesmo possa ser adaptado para outras linguagens.

## 4.8 Considerações Finais

Neste capítulo foram apresentados os principais conceitos da linha de pesquisa de mineração de aspectos. O capítulo englobou a contextualização da aplicação de aspectos em

sistemas, a apresentação das técnicas de mineração de aspectos encontradas na literatura e o aprofundamento da análise de agrupamento aplicada na mineração de aspectos, apresentando os algoritmos, medidas de distância e índices de avaliação existentes. Ao final do capítulo foram elencadas algumas limitações identificadas dos trabalhos existentes que motivaram o presente trabalho.

Com este capítulo encerra-se os fundamentos teóricos deste trabalho. No próximo capítulo é apresentada a CAAMPI, uma abordagem baseada em agrupamento para a mineração de aspectos e a identificação de pontos de corte a partir de código fonte orientado a objetos. A CAAMPI objetiva eliminar as limitações identificadas nos trabalhos existentes e aproveitar as oportunidades vislumbradas para o aperfeiçoamento da mineração de aspectos.



## CAPÍTULO 5

### CAAMPI: UMA ABORDAGEM BASEADA EM AGRUPAMENTO PARA MINERAÇÃO DE ASPECTOS E IDENTIFICAÇÃO DE PONTOS DE CORTE

Neste capítulo é apresentada a CAAMPI (*Clustering Based Approach for Aspect Mining and Pointcut Identification*), uma abordagem baseada em agrupamento para a mineração de aspectos e a identificação de pontos de corte a partir de código fonte orientado a objetos. Além disto, é descrito o CAAMPI4J, um *framework* que permite a implementação de vários tipos de instâncias da abordagem CAAMPI com base na linguagem Java.

Este capítulo está organizado nas seguintes seções: na Seção 5.1 é apresentado o processo geral da abordagem CAAMPI, na Seção 5.2 são introduzidas quatro medidas de distância, na Seção 5.3 são propostas quatro medidas de ordenação de grupos, na Seção 5.4 é descrito um método para a identificação de pontos de corte e na Seção 5.5 é descrito o *framework* CAAMPI4J.

#### 5.1 Processo CAAMPI

A abordagem CAAMPI é composta por um processo integrado que tem como entrada o código fonte do sistema orientado a objetos, e como saída, os grupos de métodos candidatos a aspectos ordenados conforme o potencial de ser ou não um aspecto e os respectivos candidatos a pontos de corte identificados. A partir da saída do processo integrado, um analista pode verificar os grupos obtidos e classificá-los através de análise manual se os mesmos são interesses transversais, ou, se tratam de interesses base ou métodos utilitários.

Os métodos utilitários, apesar de apresentarem sintomas idênticos ao de interesses transversais, como espalhamento pelos módulos e entrelaçamento dos métodos que o utilizam, não se enquadram como interesses transversais. Diferentemente de um interesse transversal, um método utilitário tem por interesse criar bibliotecas de funcionalidades de conveniência para uso essencialmente técnico, fato que desfavorece a sua refatoração para aspectos. São exemplos de funções de métodos utilitários: manipulação de cadeia de caracteres, cálculos com datas, transferência de arquivos, entre outras.

Se o grupo for classificado como interesse transversal, o analista pode verificar se o processo integrado identificou algum candidato a ponto de corte, caso positivo, o analista pode incluir o candidato a ponto de corte em sua estratégia de refatoração do interesse transversal para aspectos.

O processo geral da CAAMPI é ilustrado na Figura 5.1. Um resumo das suas contribuições é apresentado na Tabela 5.1. As suas sete fases são detalhadas a seguir:

1. **Computação:** Esta fase é composta por duas etapas. Na primeira é efetuada a análise do código fonte (*parsing*) de forma a gerar a Árvore de Sintaxe Abstrata do sistema contendo toda a hierarquia de classes e métodos. Na segunda etapa o corpo de cada método é analisado para identificar a relação de invocação entre os métodos.

Historicamente o processo de análise de código fonte exige um esforço considerável devido a complexidade das linguagens de programação. Como a abordagem CAAMPI

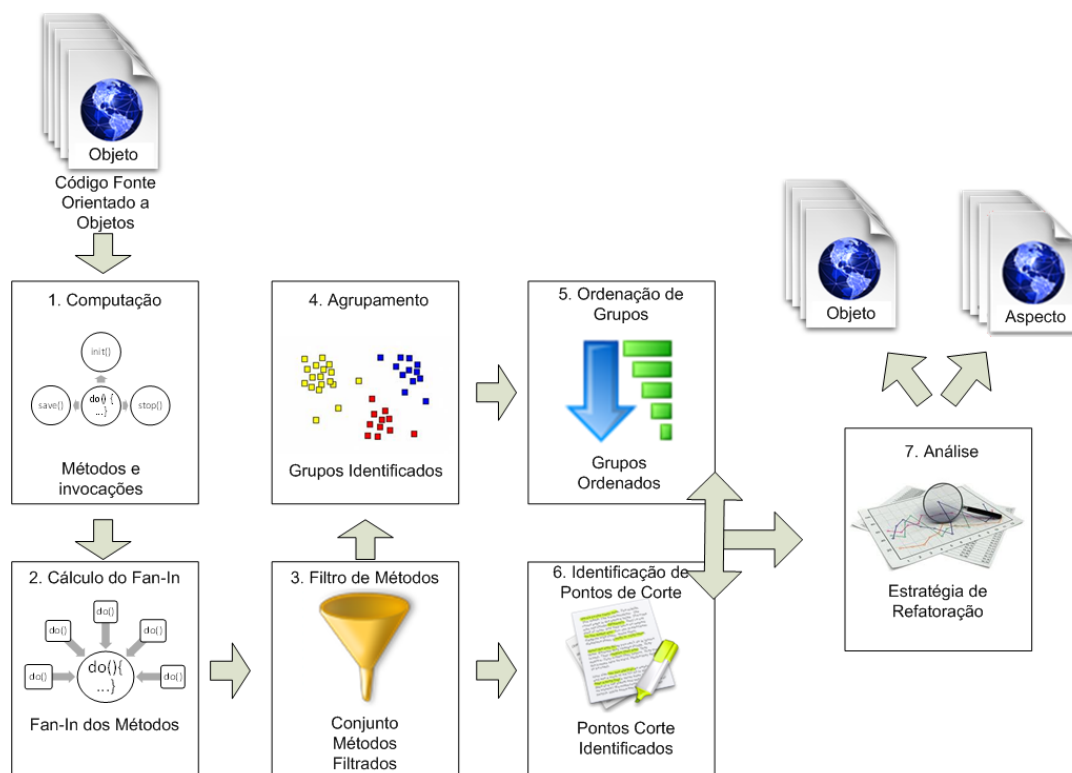


Figura 5.1: Processo da Abordagem CAAMPI

Tabela 5.1: Resumo das contribuições da CAAMPI

Fase	Contribuição
1. Computação	Fase aproveitada. Definição baseada em [57].
2. Cálculo Fan-In	Fase proposta. Cálculo do fan-in dos métodos com intuito de eliminação de interesses base.
3. Filtro de Métodos	Fase aperfeiçoada. Elimina métodos com baixo fan-in baseado no princípio de que são interesses base.
4. Agrupamento	Fase aperfeiçoada. Proposta a medida de distância $D_{SOND}$ , que combina os sintomas de espalhamento ( $D_S$ ), código duplicado ( $D_O$ ) e convenção de nomes ( $D_N$ ).
5. Ordenação de Grupos	Fase proposta. Contribuição com as medidas GSRank, GSIRank, GFRank e GFIRank que permitem o cálculo de pontuação dos grupos conforme o potencial de serem interesses transversais.
6. Pontos de Corte	Fase proposta. Contribuição com o método ARPIM, que identifica os cinco principais tipos de pontos de corte e efetua o vínculo com os grupos de candidatos a aspectos.
7. Análise	Fase aperfeiçoada. Contribuição com a disponibilização de uma pontuação para cada grupo obtido indicando o potencial do grupo participar de um interesse transversal e os respectivos pontos de corte identificados, auxiliando a análise e refatoração.

exige o conhecimento de todas as relações existentes entre os métodos, de forma que seja possível extrair as regras de associação e identificar os sintomas de interesses transversais baseados em espalhamento e entrelaçamento, este é um processo inevitável. Porém, atualmente já se encontram disponíveis gratuitamente bibliotecas e APIs que suportam a maior parte deste processo, o que reduz consideravelmente o esforço de implementação. Na Sessão 5.5 são apresentadas as APIs que foram adotadas no *framework* CAAMPI4J para suportar a análise de código fonte Java.

**Definição:** Um sistema de software  $S$  é definido como um conjunto de métodos  $M_S = \{m_1, m_2, \dots, m_n\}$ , onde  $m_i$ ,  $1 \leq i \leq n$ , é um método do sistema, e  $n$  representa o número de métodos de  $S$ .

Cada método  $m_i$  possui um conjunto de invocações (chamadas)  $C = \{c_1, c_2, \dots, c_t\}$ , onde  $c_j$ ,  $1 \leq j \leq t$ , é uma invocação efetuada pelo método  $m_i$  a outro método do sistema  $m_j \in M_S$ , e  $t$  representa a quantidade de invocações efetuadas pelo método  $m_i$ .

Dois métodos invocados  $m_1$  e  $m_2$  são considerados distintos se possuem nomes, classes, pacotes ou parâmetros distintos. Isto é,  $m_1 \neq m_2$  SE  $\text{nome}(m_1) \neq \text{nome}(m_2)$  OU  $\text{classe}(m_1) \neq \text{classe}(m_2)$  OU  $\text{pacote}(m_1) \neq \text{pacote}(m_2)$  OU  $\text{parametros}(m_1) \neq \text{parametros}(m_2)$ .

2. **Cálculo do *Fan-In*:** Nesta fase é calculado o *Fan-In* de todos os métodos. Conforme definido em [43], o *Fan-In* de um método  $m$  será igual a quantidade de corpos de métodos distintos que invocam  $m$ . E, no caso de polimorfismo, são contabilizados também os métodos distintos que invocam os métodos que  $m$  sobrescreveu e os métodos que refinaram  $m$ .
3. **Filtro de Métodos:** Nesta fase são eliminados todos os métodos e invocações que não representam interesses transversais, ou, que não contribuem para a sua identificação. São eliminados:
  - Métodos de classes internas da linguagem: Métodos internos da linguagem são eliminados. Estes métodos, por serem da própria linguagem, não são candidatos a aspectos.
  - Métodos de bibliotecas de terceiros: Métodos pertencentes a classes de biblioteca de terceiros são eliminados. Estes métodos não fazem parte do escopo do sistema em análise e portanto não são candidatos a aspectos.
  - Métodos de acesso a atributos: Métodos de acesso a atributos, também conhecidos como propriedades, são eliminados. Estes métodos não efetuam operações, já que são simplesmente métodos que encapsulam o acesso aos atributos da classe, e portanto não são candidatos a aspectos.
  - Métodos com baixo *Fan-In*: Métodos que possuem *Fan-In* inferior a um limite definido por parâmetro são eliminados. Estes métodos já possuem característica de baixo espalhamento no sistema em análise e portanto não são interessantes como candidatos a aspectos.
4. **Agrupamento:** Nesta fase é aplicado o algoritmo de agrupamento desejado para gerar os grupos de métodos. A medida de distância escolhida deverá possuir características que possibilitem a identificação de candidatos a aspectos. É sugerido o uso da medida de distância SOND (ou uma de suas derivadas), detalhada na Seção 5.2.

5. **Ordenação de Grupos:** Nesta fase é aplicada a medida de ordenação de grupos de maneira a priorizar os grupos conforme o seu potencial de ser um interesse transversal. Os grupos com maior pontuação obtida deverão ser os primeiros a serem verificados pelo analista. É sugerida a utilização de uma das medidas propostas na Seção 5.3.
6. **Identificação de Pontos de Corte:** Nesta fase são identificados os candidatos a pontos de corte. É sugerida a aplicação do método de identificação de pontos de corte ARPIM, detalhado na Seção 5.4. O ARPIM é composto de quatro etapas. Na primeira etapa, o conjunto de operações de cada método é transformado em um conjunto de transações e é aplicado um algoritmo de extração de regras de associação. Na segunda etapa as regras de associação são filtradas de forma a manter somente as regras de associação que são úteis para a identificação dos pontos de corte. Na terceira etapa as regras de associação são classificadas. Na última etapa são aplicados critérios para a determinação dos tipos de candidatos a pontos de corte identificados, encerrando com a vinculação dos candidatos a pontos de corte com os respectivos candidatos a aspectos.
7. **Análise:** Como última fase, o analista verifica os resultados obtidos após a fase de ordenação de grupos e identifica quais grupos representam interesses transversais, e quais representam interesses base ou utilitários. Quando na identificação de interesses transversais, o analista confronta os resultados com os candidatos a pontos de corte e define a estratégia para a refatoração do interesse transversal para aspectos.

## 5.2 Medida de Distância SOND

A medida de distância entre métodos SOND (*Scattering, Operations and Name Convention Distance*) tem o intuito de agregar as características de identificação dos sintomas de espalhamento, código duplicado e convenção de nomes. Portanto, se dois métodos possuem um ou mais sintomas em comum, eles devem pertencer ao mesmo grupo. Também são utilizados fatores para definir os pesos para cada um dos sintomas.

A distância SOND  $D_{SOND}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é definida na Equação 5.1:

$$D_{SOND}(m_i, m_j) = (D_S(m_i, m_j) * F_E) + (D_O(m_i, m_j) * F_O) + (D_N(m_i, m_j) * F_N) \quad (5.1)$$

Onde:

$D_S(m_i, m_j)$  : Resultado da distância de espalhamento.

$F_S$  : Fator de importância da distância de espalhamento.

$D_O(m_i, m_j)$  : Resultado da distância de operações.

$F_O$  : Fator de importância da distância de operações.

$D_N(m_i, m_j)$  : Resultado da distância de nomes.

$F_N$  : Fator de importância da distância de nomes.

Os fatores de importância definem os pesos que cada medida individual terá para compor a distância SOND, sendo que a soma de todos os fatores deve totalizar 1, ou seja,  $F_S + F_O + F_N = 1$ .

A saída da medida SOND é compreendida no intervalo entre 0 e 1, sendo:

$$\begin{cases} 0 & \text{métodos idênticos nas três medidas} \\ ]0, 1[ & \text{parcialmente dissimilares, quanto mais próximo a 0, maior a semelhança} \\ 1 & \text{métodos totalmente dissimilares nas três medidas} \end{cases}$$

As medidas de distância de espalhamento, de operações e de nomes são apresentadas com mais detalhes nas próximas subseções.

### 5.2.1 Medida de Distância de Espalhamento

O propósito da distância de espalhamento (*Scattering*) é que se dois métodos são invocados por classes e métodos em comum, ou seja, possuem um espalhamento semelhante, eles devem pertencer ao mesmo grupo.

A distância de espalhamento  $D_S(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é definida na Equação 5.2:

$$D_S(m_i, m_j) = \begin{cases} 0 & i = j \\ 1 - \frac{|Col_S(m_i) \cap Col_S(m_j)|}{|Col_S(m_i) \cup Col_S(m_j)|} & \text{se } Col_S(m_i) \cap Col_S(m_j) \neq \emptyset \\ 1 & \text{caso contrário} \end{cases} \quad (5.2)$$

Onde  $Col_S(m)$  é uma coleção que consiste de: o método  $m$ , a classe na qual o método  $m$  é definido e as classes e métodos que invocam  $m$ . Caso  $m$  possua polimorfismo, será contabilizado também o conjunto de classes e métodos que invocam os métodos que  $m$  sobrescreveu e os métodos que refinaram  $m$ . Caso existam classes aninhadas, em todos os casos será contabilizada também a classe principal que contém a classe aninhada.

A saída da medida  $D_S$  é compreendida no intervalo entre 0 e 1, sendo:

$$\begin{cases} 0 & \text{espalhamento idêntico} \\ ]0, 1[ & \text{parcialmente dissimilar, quanto mais próximo a 0, maior a semelhança} \\ 1 & \text{espalhamento totalmente dissimilar} \end{cases}$$

### 5.2.2 Medida de Distância de Operações

O propósito da distância de operações (*Operations*) é que se dois métodos efetuam operações em comum, ou seja, possuem lógica de operações semelhante ou clonada, eles devem pertencer ao mesmo grupo. Esta medida é fundamentada na questão de que alguns interesses transversais são implementados através da clonagem de código em vários módulos do sistema.

A distância de operações  $D_O(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é definida na Equação

ção 5.3:

$$D_O(m_i, m_j) = \begin{cases} 0 & \text{se } i = j \\ 1 - \frac{|Col_O(m_i) \cap Col_O(m_j)|}{|Col_O(m_i) \cup Col_O(m_j)|} & \text{se } Col_O(m_i) \cap Col_O(m_j) \neq \emptyset \\ 1 & \text{caso contrário} \end{cases} \quad (5.3)$$

Onde  $Col_O(m)$  é uma coleção que consiste das classes e métodos que  $m$  invocou e, no caso de classes aninhadas, também será contabilizada a classe principal que contém a classe aninhada.

A saída da medida  $D_O$  é compreendida no intervalo entre 0 e 1, sendo:

$$\begin{cases} 0 & \text{operações idênticas} \\ ]0, 1[ & \text{parcialmente dissimilar, quanto mais próximo a 0, maior a semelhança} \\ 1 & \text{operações totalmente dissimilares} \end{cases}$$

### 5.2.3 Medida de Distância de Nomes

O propósito da distância de nomes (*Name Convention*) é que se dois métodos possuem nomes, tipos de parâmetro e tipos de retorno em comum, eles devem pertencer ao mesmo grupo.

A distância de nomes está fundamentada no princípio de que alguns interesses transversais são implementados através do uso de convenções aplicadas ao nome de métodos e classes. Além disto, foi considerado também o fato de o método possuir o mesmo tipo de retorno e a lista de tipos de parâmetros idêntica ou com sobrecarga.

A distância de nomes  $D_N(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é definida na Equação 5.4:

$$D_N(m_i, m_j) = (D_{NM}(m_i, m_j) * F_{NM}) + (D_{NC}(m_i, m_j) * F_{NC}) + (D_{PR}(m_i, m_j) * F_{PR}) \quad (5.4)$$

Onde:

$D_{NM}(m_i, m_j)$  : Resultado da distância do nome do método.

$F_{NM}$  : Fator de importância da distância do nome do método.

$D_{NC}(m_i, m_j)$  : Resultado da distância de nome da classe.

$F_{NC}$  : Fator de importância da distância de nome da classe.

$D_{PR}(m_i, m_j)$  : Resultado da distância de parâmetros e retorno.

$F_{PR}$  : Fator de importância da distância de parâmetros e retorno.

Os fatores de importância definem os pesos que cada medida individual terá para compor a distância de nomes, sendo que a soma de todos os fatores deve totalizar 1, ou seja,  $F_{NM} + F_{NC} + F_{PR} = 1$ .

A saída da medida  $D_N$  é compreendida no intervalo entre 0 e 1, sendo:

$$\begin{cases} 0 & \text{nomes, classes, parâmetros e retornos idênticos} \\ ]0, 1[ & \text{parcialmente dissimilar, quanto mais próximo a 0, maior a semelhança} \\ 1 & \text{nomes, classes, parâmetros e retornos totalmente dissimilares} \end{cases}$$

As medidas que compõem a distância de nomes são descritas detalhadamente nos itens a seguir.

### Medida de distância de nome do método

Esta medida mede a distância de edição do nome dos métodos  $m_i$  e  $m_j$ .

Uma distância de edição mede a quantidade de edições que são necessárias para transformar uma cadeia de caracteres em outra. Entende-se por edição a necessidade de inclusão, substituição ou exclusão de caracteres. Por exemplo, para transformar a cadeia “editar” na cadeia “reedite” é necessário a inclusão dos caracteres “re” no início, a substituição do carácter “a” por “e” e a exclusão do carácter “r” no final, totalizando uma distância de 4 edições.

A distância de edição adotada para esta medida é a distância de Levenshtein [40], já que esta consegue calcular a distância de edição entre nomes de qualquer tamanho.

A distância de nomes de métodos  $D_{NM}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é definida na Equação 5.5:

$$D_{NM}(m_i, m_j) = \frac{\text{Levenshtein}_{NM}(m_i, m_j)}{\text{max}_{NM}(m_i, m_j)} \quad (5.5)$$

Onde,  $\text{Levenshtein}_{NM}(m_i, m_j)$  é a distância de edição de Levenshtein [40] entre o nome dos métodos  $m_i$  e  $m_j$  e  $\text{max}_{NM}(m_i, m_j)$  é o tamanho do nome do método mais longo entre  $m_i$  e  $m_j$ .

A saída da medida  $D_{NM}$  é compreendida no intervalo entre 0 e 1, sendo:

$$\begin{cases} 0 & \text{nome dos métodos idêntico} \\ ]0, 1[ & \text{parcialmente dissimilar, quanto mais próximo a 0, maior a semelhança} \\ 1 & \text{nome dos métodos totalmente diferente} \end{cases}$$

### Medida de distância de nome da classe

Esta medida mede especificamente a distância de edição do nome das classes a qual os métodos  $m_i$  e  $m_j$  pertencem. É considerado somente o nome da classe, excluindo o nome do pacote.

A distância de nome de classes  $D_{NC}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é definida na Equação 5.6:

$$D_{NC}(m_i, m_j) = \frac{\text{Levenshtein}_{NC}(m_i, m_j)}{\text{max}_{NC}(m_i, m_j)} \quad (5.6)$$

Onde,  $Levenshtein_{NC}(m_i, m_j)$  é a distância de edição de Levenshtein [40] entre o nome das classes dos métodos  $m_i$  e  $m_j$  e  $max_{NC}(m_i, m_j)$  é o tamanho do nome da classe mais longo entre o nome das classes de  $m_i$  e  $m_j$ .

A saída da medida  $D_{NC}$  é compreendida no intervalo entre 0 e 1, sendo:

$$\begin{cases} 0 & \text{nome das classes idêntico} \\ ]0, 1[ & \text{parcialmente dissimilar, quanto mais próximo a 0, maior a semelhança} \\ 1 & \text{nome das classes totalmente diferente} \end{cases}$$

### Medida de distância de Parâmetros e Retorno

Esta medida mede especificamente a distância entre os tipos de parâmetros e o tipo de retorno dos métodos. Entende-se por tipo, o tipo da variável declarada como parâmetro ou retorno do método.

A distância de parâmetros e retorno  $D_{PR}(m_i, m_j)$  entre dois métodos  $m_i$  e  $m_j$  é definida na Equação 5.7:

$$D_{PR}(m_i, m_j) = \frac{D_P(m_i, m_j) + D_R(m_i, m_j)}{2} \quad (5.7)$$

Sendo:

$$D_P(m_i, m_j) = \begin{cases} 0 & \text{se } Param(m_i) \cup Param(m_j) = \emptyset \\ 1 - \frac{|Param(m_i) \cap Param(m_j)|}{|Param(m_i) \cup Param(m_j)|} & \text{caso contrário} \end{cases} \quad (5.8)$$

Onde  $Param(m)$  é uma coleção que consiste dos tipos de parâmetros distintos de  $m$ , e:

$$D_R(m_i, m_j) = \begin{cases} 0 & \text{se } Retorno(m_i) = Retorno(m_j) \\ 1 & \text{se } Retorno(m_i) \neq Retorno(m_j) \end{cases} \quad (5.9)$$

Onde  $Retorno(m)$  é o tipo de retorno de  $m$ .

A saída da medida  $D_{PR}$  é compreendida no intervalo entre 0 e 1, sendo:

$$\begin{cases} 0 & \text{parâmetros e retorno dos métodos idênticos} \\ ]0, 1[ & \text{parcialmente dissimilar, quanto mais próximo a 0, maior a semelhança} \\ 1 & \text{parâmetros e retorno dos métodos totalmente diferentes} \end{cases}$$

## 5.3 Medidas de Ordenação de Grupos

A medida de ordenação de grupos tem o intuito de definir uma pontuação para cada grupo encontrado pelo algoritmo de agrupamento de forma a priorizar os grupos de acordo com o seu potencial de ser ou não um interesse transversal.



Exemplificando, para uma ordenação será necessário aplicar a medida de ordenação de grupos a todos os grupos encontrados para obter a pontuação individual. Após, os grupos serão ordenados de forma decrescente pela pontuação. Os grupos de maior pontuação terão maior potencial a serem interesses transversais do que os grupos de menor pontuação.

Nas próximas subseções são propostas quatro medidas de ordenação de grupos.

### 5.3.1 Medida de Ordenação de Grupos GSRank

A medida GSRank (*Group Scattering Ranking Measure*) tem por objetivo gerar uma pontuação a partir do espalhamento de um grupo de candidatos a aspectos. O espalhamento do grupo é dado pela quantidade de métodos distintos que invocaram os métodos pertencentes ao grupo avaliado.

Sendo  $\mathcal{K}$  o conjunto de grupos de candidatos a aspectos identificados na fase de agrupamento e  $\mathcal{K}_i$  um grupo de métodos pertencente ao conjunto  $\mathcal{K}$ . A medida de ordenação  $GSRank(\mathcal{K}_i)$  de um grupo  $\mathcal{K}_i$  é definida na Equação 5.10:

$$GSRank(\mathcal{K}_i) = \left| \bigcup_{j=1}^{|\mathcal{K}_i|} Col_S(m_j) \right| \quad (5.10)$$

Onde  $Col_S(m)$  é uma coleção composta de todos os métodos e classes que invocaram o método  $m$  e a classe a qual  $m$  pertence.

Exemplificando, o espalhamento do grupo é dado pela cardinalidade do conjunto formado pela união das coleções de métodos e classes que foram afetados pelos métodos do grupo. São considerados para a soma final somente os métodos e classes distintos. As classes as quais os métodos do grupo pertencem também são consideradas na coleção, pois, um grupo formado por métodos de classes distintas indica uma funcionalidade mais espalhada do que um grupo formado por métodos concentrados em uma única classe. O espalhamento medirá o quanto o grupo interfere no espalhamento do sistema em análise, sendo que, quanto maior o espalhamento do grupo, maior a propensão dos métodos do grupo participarem de um interesse transversal.

### 5.3.2 Medida de Ordenação de Grupos GSIRank

A medida GSIRank (*Group Scattering and Interconnectivity Ranking Measure*) tem por objetivo gerar uma pontuação a partir do produto do espalhamento do grupo obtida pela medida GSRank pela interconectividade dos métodos do grupo.

Sendo  $\mathcal{K}$  o conjunto de grupos de candidatos a aspectos identificados na fase de agrupamento e  $\mathcal{K}_i$  um grupo de métodos pertencente ao conjunto  $\mathcal{K}$ . A medida de ordenação  $GSIRank(\mathcal{K}_i)$  de um grupo  $\mathcal{K}_i$  é definida na Equação 5.11:

$$GSIRank(\mathcal{K}_i) = GSRank(\mathcal{K}_i) * interconectividade(\mathcal{K}_i) \quad (5.11)$$

Onde  $GSRank(\mathcal{K}_i)$  é o resultado da medida de ordenação GSRank e  $interconectividade(\mathcal{K}_i)$  é a interconectividade do grupo  $\mathcal{K}_i$  definida como:

$$interconectividade(\mathcal{K}_i) = 1 - \frac{1}{(|\mathcal{K}_i| * |\mathcal{K}_i|) - |\mathcal{K}_i|} \sum_{j=1}^{|\mathcal{K}_i|} \sum_{l=1}^{|\mathcal{K}_i|} \{D(m_j, m_l) | j \neq l\} \quad (5.12)$$

Onde  $D(m_j, m_l)$  é a mesma medida de distância utilizada pelo algoritmo de agrupamento.

A interconectividade (ou conectividade interna) é a média das distâncias entre todos os métodos associados ao grupo, e identifica o quanto os métodos pertencentes ao grupo são relacionados entre si. A saída é um valor no intervalo entre 0 e 1, onde 0 identifica um grupo com métodos sem nenhuma relação e 1 um grupo com métodos totalmente relacionados.

A inclusão da interconectividade do grupo na medida GSIRank tem por fundamento que grandes grupos de interesses base podem possuir um espalhamento de grupo alto, porém, os métodos do grupo terão por característica uma baixa interconectividade. De forma a penalizar os grupos de interesse base, a medida GSIRank é composta pelo produto do espalhamento do grupo pela interconectividade. Portanto, no resultado final serão priorizados os grupos que possuem os maiores níveis de espalhamento aliados a uma alta interconectividade.

### 5.3.3 Medida de Ordenação de Grupos GFRank

A medida GFRank (*Group Fan-In Ranking Measure*) tem por objetivo gerar uma pontuação a partir da soma do Fan-In dos métodos que pertencem ao grupo.

Sendo  $\mathcal{K}$  o conjunto de grupos de candidatos a aspectos identificados na fase de agrupamento e  $\mathcal{K}_i$  um grupo de métodos pertencente ao conjunto  $\mathcal{K}$ . A medida de ordenação  $GFRank(\mathcal{K}_i)$  de um grupo  $\mathcal{K}_i$  é definida na Equação 5.13:

$$GFRank(\mathcal{K}_i) = \sum_{j=1}^{|\mathcal{K}_i|} \{FanIn(m_j) | m_j \in \mathcal{K}_i\} \quad (5.13)$$

Onde  $FanIn(m)$  é o Fan-In do método  $m$ .

### 5.3.4 Medida de Ordenação de Grupos GFIRank

A medida GFIRank (*Group Fan-In and Interconnectivity Ranking Measure*) tem por objetivo gerar uma pontuação a partir do produto do resultado da medida GFRank pela interconectividade dos métodos do grupo.

Sendo  $\mathcal{K}$  o conjunto de grupos de candidatos a aspectos identificados na fase de agrupamento e  $\mathcal{K}_i$  um grupo de métodos pertencente ao conjunto  $\mathcal{K}$ . A medida de ordenação  $GFIRank(\mathcal{K}_i)$  de um grupo  $\mathcal{K}_i$  é definida na Equação 5.14:

$$GFIRank(\mathcal{K}_i) = GFRank(\mathcal{K}_i) * interconectividade(\mathcal{K}_i) \quad (5.14)$$

Onde  $GFRank(\mathcal{K}_i)$  é o resultado da medida de ordenação GFRank, e  $interconectividade(\mathcal{K}_i)$  é o mesmo índice de interconectividade utilizado pela medida GSIRank, definido na Equação 5.12.

## 5.4 ARPIM: Um método automatizado de identificação de pontos de corte

O método ARPIM (*Association Rules Based PointCut Identification Method*), método de identificação de pontos de corte baseado em regras de associação, tem por objetivo identificar automaticamente candidatos a pontos de corte a partir do código fonte do sistema.

O ARPIM possui um processo composto por quatro etapas. Na primeira etapa são extraídas as regras de associação a partir dos conjuntos de invocações dos métodos do sistema. Na segunda etapa as regras são filtradas, mantendo somente as regras úteis para a identificação de pontos de corte. Na terceira etapa as regras são classificadas em grupos conforme as suas características. Na última etapa são aplicados critérios sobre as regras remanescentes para a determinação dos 5 tipos de candidatos a pontos de corte: Antes da Chamada, Depois da Chamada, Antes da Execução, Durante a Execução e Após a Execução.

As etapas são detalhadas a seguir.

### 1. Extração de Regras de Associação

Nesta fase é aplicado um algoritmo de mineração de regras de associação para encontrar regras de associação a partir do conjunto de invocações dos métodos.

Conceitualizando, o objetivo da mineração de regras de associação [1] [38] é encontrar um conjunto de regras de associação a partir de um conjunto de itens e transações. Um conjunto de itens  $I = \{i_1, i_2, \dots, i_m\}$  é formado pelos elementos que serão relacionados e o conjunto de transações  $T = \{t_1, t_2, \dots, t_n\}$  é formado pelas transações onde ocorrem as relações entre os itens, sendo que cada transação  $t_i$  é formada por um subconjunto de itens, ou seja  $t_i \subseteq I$ . É uma regra de associação é uma implicação na forma  $X \Rightarrow Y$ , onde  $X$  é a pré condição (antecedente) e  $Y$  é a pós condição (consequente), sendo  $X$  e  $Y$  composto por itens distintos, ou seja  $X, Y \subset I$ ,  $X \neq \emptyset$ ,  $Y \neq \emptyset$  e  $X \cap Y = \emptyset$ . Tanto a pré condição quanto a pós condição de uma regra podem conter um ou mais itens, sendo a quantidade de itens chamada de comprimento do conjunto (*k-itemsets*).

Existem dois parâmetros básicos que são utilizados pelos algoritmos de mineração de regras de associação [38]: o suporte e a confiança. O suporte define a porcentagem mínima de ocorrências de  $X$  e  $Y$  em relação ao número total de transações que deverão existir para que uma regra seja considerada. Já a confiança define a porcentagem mínima de transações que contêm  $X \cup Y$  em relação ao número total de transações que contêm  $X$ , indicando a probabilidade mínima da implicação de  $Y$  quando na ocorrência de  $X$ ,  $P(Y|X)$ .

Exemplo: Considerando 100 transações e restrições de suporte de 10% e confiança de 80%. Serão extraídas somente as regras em que  $X \cup Y$  exceda a  $100 * 10\% = 10$  ocorrências, e, em que mais de 80% das transações que contêm  $X$ ,  $X$  implique em  $Y$ .

No ARPIM, o conjunto de itens  $I$  é constituído pelos  $m$  métodos do sistema  $S$  e é estendido com dois métodos fictícios, `inicio()` e `fim()`, que foram introduzidos para identificar o início e o fim de cada uma das transações. Portanto o conjunto de itens possui o formato  $I = \{\text{inicio}(), i_1, i_2, \dots, i_m, \text{fim}()\}$ . Já o conjunto de transação é formado a partir das invocações efetuadas pelos  $m$  métodos do sistema,

denominadas operações. Por padrão a transação de um método  $m$  é composto pelo conjunto de invocações (chamadas)  $C = \{c_1, c_2, \dots, c_l\}$ , onde  $c_j$  é uma invocação efetuada pelo método  $m_i$  a outro método do sistema e  $l$  é a quantidade de invocações efetuadas pelo método  $m_i$ . A transação é estendida do item `inicio()` como a primeira operação e do item `fim()` como a última operação. Portanto uma transação de um método  $m$  possui o formato  $t_i = \{inicio(), i_1, i_2, \dots, i_l, fim()\}$ .

Após a formação dos conjuntos de itens e transações, é aplicado o algoritmo de mineração de regras de associação escolhido para a extração do conjunto de regras. O algoritmo selecionado deverá ser configurado para extrair o conjunto de itens com comprimento dois (*2-itemset*), com suporte mínimo de  $2/m$ , duas ocorrências em relação ao total de transações (métodos), e confiança de 75%. Ou seja, o algoritmo deverá extrair regras com somente um item de pré condição e um item de pós condição ( $i_x \Rightarrow i_y$ ), com ao menos duas ocorrências de  $i_x$  e  $i_y$  e probabilidade de que  $i_x$  implica em  $i_y$  maior ou igual a 75%,  $P(i_y|i_x) \geq 75\%$ . É sugerida a utilização do Apriori [1] ou outro algoritmo que suporte estes parâmetros.

## 2. Filtro de Regras

O resultado da fase de extração de regras de associação é composto por regras que envolvem métodos (itens) que participam em várias transações, podendo existir tanto regras de associação entre operações adjuntas, invocadas em sequência, quanto disjuntas, invocadas dentro dos mesmos métodos entrelaçados, porém, não sequencialmente.

Devido ao fato de que os critérios utilizados para a identificação dos pontos de corte dependerem também da informação da relação de sequência e antecedência de operações, regras envolvendo métodos disjuntos, ou seja, cujas operações não ocorrem em sequência, devem ser eliminadas.

É qualificada como regra de associação de métodos adjuntos a regra cuja pré condição (antecedente) é seguida ou antecedida da pós condição (consequente) nos métodos entrelaçados. Ou seja, são mantidas somente as regras de associação que possuem um dos seguintes formatos:  $SE\ op_n \Rightarrow op_{n+1}$  ou  $SE\ op_n \Rightarrow op_{n-1}$ .

Na Tabela 5.2 é apresentado um exemplo didático de como são filtradas as regras entre métodos adjuntos.

Tabela 5.2: Exemplo de filtro de regras entre métodos adjuntos

Transações	Regras Válidas	Regras Eliminadas
$t_1 = \{inicio, a, b, fim\}$ $t_2 = \{inicio, a, b, fim\}$	$SE\ inicio \Rightarrow a$	$SE\ inicio \Rightarrow b$
	$SE\ a \Rightarrow inicio$	$SE\ inicio \Rightarrow fim$
	$SE\ a \Rightarrow b$	$SE\ a \Rightarrow fim$
	$SE\ b \Rightarrow a$	$SE\ b \Rightarrow inicio$
	$SE\ b \Rightarrow fim$	$SE\ fim \Rightarrow inicio$
	$SE\ fim \Rightarrow b$	$SE\ fim \Rightarrow a$

## 3. Classificação de Regras

Esta fase tem por objetivo refinar o resultado da extração de regras classificando as regras de associação em quatro grupos mutuamente exclusivos:

- Regra de Início de Execução: É classificada como regra de início de execução todas as regras de associação no formato SE  $op_n \Rightarrow inicio()$ . Esta regra indica que  $op_n$  ocorre sempre como a primeira operação nos métodos entrelaçados. Regras no formato SE  $inicio() \Rightarrow op_n$  não foram consideradas. Para que esta situação ocorra, a primeira operação de todo e qualquer método do sistema deve ser sempre  $op_n$ , fato que não ocorre na prática.
- Regra de Fim de Execução: É classificada como regra de fim de execução todas as regras de associação no formato SE  $op_n \Rightarrow fim()$ . Esta regra indica que  $op_n$  ocorre sempre como a última operação nos métodos entrelaçados. Regras no formato SE  $fim() \Rightarrow op_n$  não foram consideradas. Para que esta situação ocorra, a última operação de todo e qualquer método do sistema deve ser sempre  $op_n$ , fato que não ocorre na prática.
- Regra de Sequência de Chamada: É classificada como regra de sequência de chamada todas as regras de associação no formato SE  $op_n \Rightarrow op_{n+1}$ . Esta regra indica que a operação  $op_n$  é sempre seguida da invocação a operação  $op_{n+1}$  nos métodos entrelaçados.
- Regra de Antecedência de Chamada: É classificada como regra de antecedência de chamada todas as regras de associação no formato SE  $op_n \Rightarrow op_{n-1}$ . Esta regra indica que a operação  $op_n$  é sempre antecedida da invocação a operação  $op_{n-1}$  nos métodos entrelaçados.

Na Tabela 5.3 é apresentado um exemplo didático de como são classificadas as regras obtidas.

Tabela 5.3: Exemplo de classificação de regras

Transações	Regras Válidas	Regras classificadas
$t_1 = \{inicio, a, b, fim\}$ $t_2 = \{inicio, a, b, fim\}$	SE $inicio \Rightarrow a$	Desconsiderada
	SE $a \Rightarrow inicio$	$a$ INÍCIO EXECUÇÃO $[t_1, t_2]$
	SE $a \Rightarrow b$	$a$ SEGUIDO $b$
	SE $b \Rightarrow a$	$b$ ANTECEDIDO $a$
	SE $b \Rightarrow fim$	$b$ FIM EXECUÇÃO $[t_1, t_2]$
	SE $fim \Rightarrow b$	Desconsiderada

#### 4. Critérios de identificação de Candidatos a Ponto de Corte

Esta fase tem por objetivo aplicar critérios para a identificação de candidatos a ponto de corte a partir das regras de associação classificadas e vincular os resultados aos grupos de candidatos a aspectos identificados pela técnica de agrupamento.

São aplicados os seguintes critérios para a identificação de cada um dos tipos de pontos de corte:

- Antes da Execução: Sempre que existir uma regra de associação classificada como Início de Execução em que a pré-condição é o método  $m$ , será considerado que  $m$  é um candidato a ponto de corte antes da execução. É considerado o princípio de que esta situação pode ser refatorada através de um ponto de corte antes da execução dos métodos entrecortados. Exemplificando:

$$t_1 = \{inicio, m, \dots\}, \dots, t_9 = \{inicio, m, \dots\}$$

$$m \text{ INÍCIO EXECUÇÃO } [t_1, \dots, t_9] \Rightarrow PC \ m \text{ ANTES EXECUÇÃO } [t_1, \dots, t_9]$$

- Após a Execução: Sempre que existir uma regra de associação classificada como Fim de Execução em que a pré-condição é o método  $m$ , será considerado que  $m$  é um candidato a ponto de corte após a execução. Exemplificando:

$$t_1 = \{\dots, m, fim\}, \dots, t_9 = \{\dots, m, fim\}$$

$$m \text{ FIM EXECUÇÃO } [t_1, \dots, t_9] \Rightarrow \text{PC } m \text{ APÓS EXECUÇÃO } [t_1, \dots, t_9]$$

- Durante a Execução: Sempre que coexistirem duas regras, uma classificada como Sequência da Chamada em que a pré condição é um método  $x$  e a pós condição é o método candidato a aspecto  $m$ , e outra classificada como Antecedência da Chamada em que a pré condição é o mesmo método  $x$  e a pós condição é o método candidato a aspecto  $m$ , será considerado que  $m$  é um candidato a ponto de corte durante a execução de  $x$ . Exemplificando:

$$t_1 = \{\dots, m, x, m, \dots\}, t_2 = \{\dots, m, x, m, \dots\}$$

$$\begin{cases} x \text{ SEGUIDO } m \\ x \text{ ANTECEDIDO } m \end{cases} \Rightarrow \text{PC } m \text{ DURANTE EXECUÇÃO } x$$

- Antes da Chamada: Sempre que existir uma regra de associação classificada como Antecedência da Chamada em que a pré-condição é o método  $x$  e a pós condição é o método candidato a aspecto  $m$ , será considerado que  $m$  é um candidato a ponto de corte antes da chamada do método  $x$ . Exemplificando:

$$t_1 = \{\dots, m, x, \dots\}, t_2 = \{\dots, m, x, \dots\}$$

$$x \text{ ANTECEDIDO } m \Rightarrow \text{PC } m \text{ ANTES CHAMADA } x$$

- Após a Chamada: Sempre que existir uma regra de associação classificada como Sequência da Chamada em que a pré-condição é o método  $x$  e a pós condição é o método candidato a aspecto  $m$ , será considerado que  $m$  é um candidato a ponto de corte após a chamada do método  $x$ . Exemplificando:

$$t_1 = \{\dots, x, m, \dots\}, t_2 = \{\dots, x, m, \dots\}$$

$$x \text{ SEGUIDO } m \Rightarrow \text{PC } m \text{ APÓS CHAMADA } x$$

Ao final, os pontos de corte identificados são agrupados por método e vinculados aos respectivos grupos de candidatos a aspectos obtidos na fase de agrupamento. Exemplificando, dado um grupo  $G_1$  obtido pelo algoritmo de agrupamento, composto pelos métodos  $a$ ,  $b$  e  $c$ . Todos os pontos de corte identificados referentes a métodos pertencentes ao grupo  $G_1$  serão vinculados a este. Exemplo:

$$\begin{cases} \text{PC } a \text{ DURANTE EXECUÇÃO } z \\ \text{PC } b \text{ APÓS CHAMADA } w \\ \text{PC } b \text{ APÓS CHAMADA } k \\ \text{PC } c \text{ ANTES CHAMADA } y \end{cases} \Rightarrow G_1\{a, b, c\}$$

## 5.5 CAAMPI4J: Um *framework* de implementação da CAAMPI para Java

De forma a demonstrar a aplicabilidade da abordagem CAAMPI, foi desenvolvido o CAAMPI4J, um *framework* que permite a implementação de vários tipos de instâncias da abordagem CAAMPI em linguagem Java.

O CAAMPI4J define as características técnicas necessárias para permitir a criação de uma ferramenta automatizada que possibilita a análise de código fonte escrito em Java. Como resultado do *framework* foi gerada uma ferramenta toda implementada em Java, sendo a única exceção uma interface através de troca de arquivos com o programa de linha de comando hMETIS, utilizado quando o algoritmo de agrupamento escolhido é o CHAMELEON.

A ferramenta é parametrizável e permite a escolha: do pacote do sistema em Java que será analisado; do algoritmo de agrupamento; da medida de distância; da medida de ordenação; e dos parâmetros específicos de cada algoritmo.

As características técnicas de implementação das fases da abordagem CAAMPI no *framework* CAAMPI4J são detalhadas nas próximas subseções.

### 1. Computação

Na abordagem CAAMPI, a fase de computação é composta por duas etapas:

#### Extração da Árvore de Sintaxe Abstrata

Nesta etapa é extraída a árvore de sintaxe abstrata do sistema, contendo toda a hierarquia de classes, atributos e métodos.

No CAAMPI4J a análise do código fonte (*parsing*) foi implementada com o uso da biblioteca *Compiler Tree API* [48]. A *Compiler Tree API* é uma biblioteca interna da JDK 1.6 (*Java Development Kit*) que possibilita a extração da árvore de sintaxe abstrata a partir do código fonte Java. A vantagem desta biblioteca é que ela utiliza o próprio compilador da plataforma Java, o que elimina a ocorrência de erros de *parsing*.

#### Extração da Relação de Invocação entre os Métodos

Nesta etapa o corpo dos métodos identificados na árvore de sintaxe abstrata são analisados e deles são extraídos todas as invocações efetuadas. As invocações têm suas classes e métodos identificados, e, quando do próprio sistema, é criado um vínculo entre o método invocador e o método invocado.

No CAAMPI4J a extração das invocações do corpo dos métodos foi implementada com auxílio da biblioteca *Compiler Tree API* [48]. O contexto de atributos de classe, parâmetros dos métodos e variáveis locais é armazenado. O armazenamento do contexto permite a identificação da classe do método invocado através do tipo de variável ou estrutura de dados acessada. Após a identificação da classe, é utilizada a *Reflection API* [49], biblioteca interna da plataforma Java, para buscar o método exato que foi invocado, comparando o nome e tipos de parâmetros utilizados.

O modelo de classes utilizado para armazenamento de todas as informações do processo de computação é representado na Figura 5.2. Este modelo permite a navegação tanto de cima para baixo (*top-down*) quanto de baixo para cima (*bottom-up*) pela estrutura de classes, métodos e invocações, incluindo a





## 2. Cálculo do Fan-in

A partir das informações obtidas na fase computação, o CAAMPI4J calcula o *Fan-In* de todos os métodos conforme definido em [43]. O valor do *Fan-In* é anexado as informações do método. Caso o analista deseje, o CAAMPI4J permite que seja extraída a listagem de todos os métodos com o respectivo *Fan-In*, possibilitando a identificação de métodos com alto *Fan-In*. No Apêndice C é apresentada a lista de métodos que foi extraída a partir dos sistemas JHotDraw, Tomcat e HSQLDB.

## 3. Filtro de Métodos

Nesta fase a abordagem CAAMPI aplica um filtro para eliminar os métodos que não representam interesses transversais.

No CAAMPI4J são eliminados:

- Métodos de classes internas da linguagem Java: Métodos internos, por exemplo `equals`, `hashCode`, `clone` e `toString` são eliminados. Estes métodos, por serem da própria plataforma Java, não são candidatos a aspectos.
- Métodos de bibliotecas de terceiros: Métodos pertencentes a classes de biblioteca de terceiros são eliminados. Estes métodos não fazem parte do escopo do sistema em análise e portanto não são candidatos a aspectos.
- Métodos de acesso a variáveis: Métodos de acesso a variáveis, com prefixo *get* e *set*, são eliminados. Estes métodos não efetuam operações, já que são simplesmente métodos que encapsulam o acesso aos atributos da classe, e portanto não são candidatos a aspectos.
- Métodos com baixo *Fan-In*: Métodos que possuem *Fan-In* inferior a 2 são eliminados. O parâmetro 2 foi escolhido como padrão (*default*) devido ao fato de ser conservador, já que métodos que não possuem nenhuma invocação, ou, que foram invocados de somente um local no sistema, não possuem sintoma de espalhamento. Outros valores também podem ser configurados conforme a necessidade, sendo que quanto maior o valor, maior a quantidade de métodos eliminados, o que reduz o tempo de processamento, porém, introduz o risco de eliminar métodos importantes para a identificação de candidatos a aspectos.

No final é obtido o conjunto de métodos filtrado que será utilizado como base para as fases de agrupamento e identificação de pontos de corte.

## 4. Agrupamento

Nesta fase a abordagem CAAMPI aplica o algoritmo de agrupamento para gerar grupos de métodos candidatos a aspectos.

No CAAMPI4J foram implementados três algoritmos de agrupamento, seis medidas de distância e um algoritmo para determinação dos grupos iniciais, que serão apresentados a seguir.

### Determinação de grupos iniciais

Para os algoritmos de agrupamento que dependem da informação prévia dos grupos iniciais ou da quantidade de grupos o CAAMPI4J utiliza o algoritmo de seleção de métodos representativos, adaptado dos trabalhos [57] [12] [56] e [13], conforme esboçado no Algoritmo 3 e detalhado a seguir:

1. O método escolhido como o primeiro método representativo é o método mais distante do conjunto de todos os métodos (o método que maximiza a soma das distâncias de todos os outros métodos).
2. Para cada um dos métodos remanescentes (que não foram escolhidos), é computada a distância mínima ( $dmin$ ) entre o método e os métodos representativos escolhidos. O próximo método representativo a ser escolhido é o método  $m$  que maximiza  $dmin$  e cuja distância seja maior que o limite  $distMin$ . Se tal método não existir, o processo é finalizado, caso contrário, o método selecionado é considerado como um novo método representativo e o processo é repetido novamente.

<p><b>Entrada:</b> <math>M</math>: Métodos do sistema após a fase de filtro</p> <p><b>Saída:</b> <math>R</math>: Métodos representativos</p> <pre> 1 início 2   <math>R_1 \leftarrow \max_{i=1,  M } \left\{ \sum_{j=1, j \neq i}^{ M } D(m_i, m_j) \right\};</math> 3   <math>r \leftarrow 1</math>; // Quantidade métodos escolhidos 4   enquanto <i>Houver novos métodos representativos faça</i> 5     <math>E \leftarrow \{j   j \notin \{R_1, \dots, R_r\}, d = \min_{l=1, r} \{D(m_j, m_{R_l})\}, d &gt; distMIN\};</math> 6     se <math>E = \emptyset</math> então 7       // Critério de parada atingido 8     senão 9       <math>r \leftarrow r + 1</math>; // Método representativo escolhido 10      <math>R_r \leftarrow \max_{j \in E} \{ \min_{l=1, r-1} \{d(m_j, m_{R_l})\} \};</math> 11    fim 12 fim</pre>
---

**Algoritmo 3:** Algoritmo de seleção de métodos representativos.

### Algoritmos de Agrupamento

Os três algoritmos de agrupamento implementados no CAAMPI4J são:

- *k-medoids*: Foi implementado o algoritmo *k-medoids* clássico conforme descrito na Seção 3.3. Os *medoids* iniciais são os métodos identificados pelo algoritmo de seleção de métodos representativos conforme esboçado no Algoritmo 3. O *k-medoids* é executado até que não haja mais troca de *medoids*.
- Hierárquico aglomerativo clássico: Foi implementado o algoritmo hierárquico aglomerativo clássico conforme descrito na Seção 3.3. O método de ligação escolhido foi o *complete linkage*, onde a distância entre um par de grupos é determinada pelos dois objetos mais distantes entre os grupos. O critério de parada aplicado foi a aglomeração de grupos até que a quantidade de grupos gerados seja igual a quantidade de métodos identificados pelo algoritmo de seleção de métodos representativos, conforme esboçado no Algoritmo 3.
- CHAMELEON: Foi implementado o algoritmo CHAMELEON conforme descrito no estudo original [33]. Do mesmo modo que em [33], foi utilizado o aplicativo hMETIS [28] para efetuar o particionamento dos hipergrafos.

A interface entre o Java e o hMETIS foi efetuada através da geração do arquivo de hipergrafo no formato esperado pelo hMETIS e a carga do arquivo de retorno gerado pelo hMETIS com a partição efetuada.

### Medidas de distância

As seis medidas de distância implementadas no CAAMPI4J são:

- Distância  $D_S^{CCC}$ : Medida de distância de espalhamento utilizada pelos autores Czibula, Cojocar e Czibula nos estudos [12] e [13] (Seção 4.4.3.3).
- Distância  $D^{HBZH}$ : Medida de distância de espalhamento utilizada pelos autores He, Bai, Zhang e Hu no estudo [27] (Seção 4.4.3.5).
- Distância  $D_S$ : Medida de distância de espalhamento proposta neste trabalho (Seção 5.2.1).
- Distância  $D_O$ : Medida de distância de operações proposta neste trabalho (Seção 5.2.2).
- Distância  $D_N$ : Medida de distância de nomes proposta neste trabalho (Seção 5.2.3).
- Distância  $D_{SOND}$ : Medida de distância proposta neste trabalho resultante da combinação das medidas de espalhamento, operações e nomes (Seção 5.2).

## 5. Ordenação de Grupos

Nesta fase a abordagem CAAMPI aplica a medida de ordenação de grupos de maneira a priorizar os grupos conforme o seu potencial de ser ou não um interesse transversal. Para isto, é calculada a pontuação para todos os grupos obtidos na fase de agrupamento. Ao final os grupos são ordenados em ordem decrescente de pontuação.

No CAAMPI4J foram implementadas as medidas de ordenação de grupos GSRank, GSIRank, GFRank e GFIRank conforme detalhadas na Seção 5.3.

## 6. Identificação de Pontos de Corte

O CAAMPI4J implementa o método de identificação de pontos de corte ARPIM conforme detalhado na Seção 5.4. Para a geração do conjunto de itens é criada uma identificação numérica única para cada um dos métodos remanescentes após a fase de filtro de métodos. Esta identificação é propositalmente a mesma identificação do método utilizado na fase de agrupamento, ou seja,  $m_1 \Rightarrow i_1$ . Para o item fictício `inicio()` é atribuído o identificador 0 ( $i_0$ ) e para o item fictício `fim()` é atribuído o valor máximo admitido para um número inteiro ( $i_{MAX}$ ).

Para a geração do conjunto de transações são identificadas todas as invocações efetuadas por cada um dos métodos, vinculando cada método invocado com o seu respectivo identificador de item. Sendo que cada transação foi estendida com o item  $i_0$  como primeira operação e o item  $i_{MAX}$  como última operação.

Por exemplo, dado um sistema que possua dez métodos, é formado um conjunto de itens  $I = \{i_0, i_1, \dots, i_{10}, i_{MAX}\}$ . Destes métodos, serão tomados os métodos  $m_1$  a  $m_4$  para extração do conjunto de transações  $T = \{t_1, \dots, t_4\}$ , sendo:

$$m_1() = t_1\{i_0, i_3, i_5, i_6, i_5, i_7, i_{MAX}\}$$

$$m_2() = t_2\{i_0, i_5, i_6, i_{MAX}\}$$

$$m_3() = t_3\{i_0, i_1, i_2, i_5, i_6, i_5, i_7, i_{MAX}\}$$

$$m_4() = t_4\{i_0, i_1, i_5, i_6, i_5, i_{MAX}\}$$

A partir do momento que conjunto de itens  $I$  e transações  $T$  é formado, é aplicado o algoritmo de extração de regras de associação Apriori [1], configurado conforme definição na Seção 5.4. Considerando o conjunto de transações  $T$ , o Apriori obtêm um conjunto de vinte regras de associação, conforme listado na primeira coluna da Tabela 5.4.

Na fase de filtro, para cada regra obtida pelo Apriori, é validada se a pré condição possui uma relação de sequência e/ou antecedência com a pós condição. Caso seja confirmada que a relação atende aos critérios de suporte mínimo de 2 ocorrências e confiança de 75%, a relação é aceita, senão é rejeitada. Na Tabela 5.4 são listadas todas as relações alvo de validação, a quantidade de ocorrências, a confiança e se a relação foi aceita ou não.

Tabela 5.4: Regras Obtidas no Exemplo

Regra obtida Apriori	Relação a Validar	Ocorrências	Confidência	Aceitação
SE $i_0$ ENTAO $i_{MAX}$	$i_0$ ANTECEDIDO $i_{MAX}$	0	0/4 = 0%	Rejeitada
	$i_0$ SEGUIDO $i_{MAX}$	0	0/4 = 0%	Rejeitada
SE $i_0$ ENTAO $i_5$	$i_0$ ANTECEDIDO $i_5$	0	0/4 = 0%	Rejeitada
	$i_0$ SEGUIDO $i_5$	1	1/4 = 25%	Rejeitada
SE $i_0$ ENTAO $i_6$	$i_0$ ANTECEDIDO $i_6$	0	0/4 = 0%	Rejeitada
	$i_0$ SEGUIDO $i_6$	0	0/4 = 0%	Rejeitada
SE $i_1$ ENTAO $i_0$	$i_1$ ANTECEDIDO $i_0$	2	2/2 = 100%	<u>Aceita</u>
	$i_1$ SEGUIDO $i_0$	0	0/2 = 0%	Rejeitada
SE $i_1$ ENTAO $i_5$	$i_1$ ANTECEDIDO $i_5$	0	0/2 = 0%	Rejeitada
	$i_1$ SEGUIDO $i_5$	1	1/2 = 50%	Rejeitada
SE $i_1$ ENTAO $i_6$	$i_1$ ANTECEDIDO $i_6$	0	0/2 = 0%	Rejeitada
	$i_1$ SEGUIDO $i_6$	0	0/2 = 0%	Rejeitada
SE $i_1$ ENTAO $i_{MAX}$	$i_1$ ANTECEDIDO $i_{MAX}$	0	0/2 = 0%	Rejeitada
	$i_1$ SEGUIDO $i_{MAX}$	0	0/2 = 0%	Rejeitada
SE $i_5$ ENTAO $i_0$	$i_5$ ANTECEDIDO $i_0$	1	1/7 = 14%	Rejeitada
	$i_5$ SEGUIDO $i_0$	0	0/7 = 0%	Rejeitada
SE $i_5$ ENTAO $i_6$	$i_5$ ANTECEDIDO $i_6$	3	3/7 = 42%	Rejeitada
	$i_5$ SEGUIDO $i_6$	4	4/7 = 57%	Rejeitada
SE $i_5$ ENTAO $i_{MAX}$	$i_5$ ANTECEDIDO $i_{MAX}$	0	0/7 = 0%	Rejeitada
	$i_5$ SEGUIDO $i_{MAX}$	1	1/7 = 14%	Rejeitada
SE $i_6$ ENTAO $i_0$	$i_6$ ANTECEDIDO $i_0$	0	0/4 = 0%	Rejeitada
	$i_6$ SEGUIDO $i_0$	0	0/4 = 0%	Rejeitada
SE $i_6$ ENTAO $i_5$	$i_6$ ANTECEDIDO $i_5$	4	4/4 = 100%	<u>Aceita</u>
	$i_6$ SEGUIDO $i_5$	3	3/4 = 75%	<u>Aceita</u>
SE $i_6$ ENTAO $i_{MAX}$	$i_6$ ANTECEDIDO $i_{MAX}$	0	0/4 = 0%	Rejeitada
	$i_6$ SEGUIDO $i_{MAX}$	1	1/4 = 25%	Rejeitada
SE $i_7$ ENTAO $i_0$	$i_7$ ANTECEDIDO $i_0$	0	0/2 = 0%	Rejeitada
	$i_7$ SEGUIDO $i_0$	0	0/2 = 0%	Rejeitada
SE $i_7$ ENTAO $i_5$	$i_7$ ANTECEDIDO $i_5$	2	2/2 = 100%	<u>Aceita</u>
	$i_7$ SEGUIDO $i_5$	0	0/2 = 0%	Rejeitada
SE $i_7$ ENTAO $i_6$	$i_7$ ANTECEDIDO $i_6$	0	0/2 = 0%	Rejeitada
	$i_7$ SEGUIDO $i_6$	0	0/2 = 0%	Rejeitada
SE $i_7$ ENTAO $i_{MAX}$	$i_7$ ANTECEDIDO $i_{MAX}$	0	0/2 = 0%	Rejeitada
	$i_7$ SEGUIDO $i_{MAX}$	2	2/2 = 100%	<u>Aceita</u>
SE $i_{MAX}$ ENTAO $i_0$	$i_{MAX}$ ANTECEDIDO $i_0$	0	0/4 = 0%	Rejeitada
	$i_{MAX}$ SEGUIDO $i_0$	0	0/4 = 0%	Rejeitada

Continua na próxima página ...

.. continuação da página anterior				
Regra Apriori	Relação a Validar	Ocorrências	Confidência	Aceitação
SE $i_{MAX}$ ENTAO $i_5$	$i_{MAX}$ ANTECEDIDO $i_5$	1	1/4 = 25%	Rejeitada
	$i_{MAX}$ SEGUIDO $i_5$	0	0/4 = 0%	Rejeitada
SE $i_{MAX}$ ENTAO $i_6$	$i_{MAX}$ ANTECEDIDO $i_6$	1	1/4 = 25%	Rejeitada
	$i_{MAX}$ SEGUIDO $i_6$	0	0/4 = 0%	Rejeitada

No exemplo foram aceitas somente cinco relações, conforme listadas na primeira coluna da Tabela 5.5. O próximo passo é a classificação das regras conforme o tipo de relação aceita em um dos 4 tipos: Antecedência de Chamada, Sequência de Chamada, Início de Execução e Fim de Execução. A classificação final das regras é apresentada na Tabela 5.5.

Tabela 5.5: Classificação das relações encontradas

Relação Aceita	Regra Classificada
$i_1$ ANTECEDIDO $i_0$	$i_1$ INÍCIO EXECUÇÃO [ $m_3()$ , $m_4()$ ]
$i_6$ ANTECEDIDO $i_5$	$i_6$ ANTECEDIDO CHAMADA $i_5$
$i_6$ SEGUIDO $i_5$	$i_6$ SEGUIDO CHAMADA $i_5$
$i_7$ ANTECEDIDO $i_5$	$i_7$ ANTECEDIDO CHAMADA $i_5$
$i_7$ SEGUIDO $i_{MAX}$	$i_7$ FIM EXECUÇÃO [ $m_1()$ , $m_3()$ ]

A partir das regras classificadas listadas na segunda coluna da Tabela 5.5, são aplicados os critérios para a determinação do tipo de candidato a ponto de corte encontrado. O mapeamento das regras classificadas com os pontos de corte identificados é apresentado na Tabela 5.6.

Tabela 5.6: Candidatos a Pontos de Corte Identificados

Regra Classificada	Ponto de Corte Identificado
$i_1$ INÍCIO EXECUÇÃO [ $m_3()$ , $m_4()$ ]	$i_1$ ANTES DA EXECUÇÃO [ $m_3()$ , $m_4()$ ]
$i_6$ ANTECEDIDO CHAMADA $i_5$ e $i_6$ SEGUIDO CHAMADA $i_5$	$i_5$ DURANTE A EXECUÇÃO $i_6$
$i_7$ ANTECEDIDO CHAMADA $i_5$	$i_5$ ANTES DA CHAMADA $i_7$
$i_7$ FIM EXECUÇÃO [ $m_1()$ , $m_3()$ ]	$i_7$ APÓS A EXECUÇÃO [ $m_1()$ , $m_3()$ ]

De posse dos pontos de corte listados na segunda coluna da Tabela 5.6, é necessário efetuar o vínculo destes candidatos a pontos de corte com os grupos de candidatos a aspectos identificados na fase de agrupamento. Primeiramente, é necessário recuperar os métodos aos quais os itens se referem, e, agrupar os pontos de corte por método. Como inicialmente o identificador do item foi definido propositalmente idêntico ao identificador do método da fase de agrupamento, a recuperação do método é efetuada através da simples conciliação do identificador do item com o identificador do método, ou seja,  $i_x \Rightarrow m_x$ . Os candidatos a pontos de corte agrupados por métodos são listados na Tabela 5.7.

Tabela 5.7: Candidatos a Pontos de Corte Agrupados por Método

Método	Ponto de Corte
$m_1$	$m_1$ ANTES DA EXECUÇÃO [ $mC()$ , $mD()$ ]
$m_5$	$m_5$ DURANTE A EXECUÇÃO $m_6$ , $m_5$ ANTES DA CHAMADA $m_7$
$m_7$	$m_7$ APÓS A EXECUÇÃO [ $mA()$ , $mC()$ ]

Por último, é feita a vinculação dos candidatos a pontos de corte com os respectivos grupos de candidatos a aspectos que contêm os métodos aos quais os pontos de corte se referem. Supondo que os grupos encontrados pela técnica de agrupamento sejam:

$$G_1\{m_3, m_4\}, G_2\{m_1\}, G_3\{m_5\}, G_4\{m_6\} \text{ e } G_5\{m_7\}.$$

O resultado é representado na Tabela 5.8. Na primeira coluna é apresentado o grupo encontrado na fase de agrupamento, na segunda coluna os métodos que compõem o grupo e na última coluna os candidatos a pontos de corte, que foram vinculados conforme os métodos que pertencem a cada grupo.

Tabela 5.8: Vinculação dos Pontos de Corte aos Grupos

Grupo	Métodos	Candidatos a Pontos de Corte
$G_1$	$m_3, m_4$	$\emptyset$
$G_2$	$m_1$	$m_1$ ANTES DA EXECUÇÃO [mC(),mD()]
$G_3$	$m_5$	$m_5$ DURANTE A EXECUÇÃO $m_6, m_5$ ANTES DA CHAMADA $m_7$
$G_4$	$m_6$	$\emptyset$
$G_5$	$m_7$	$m_7$ APÓS A EXECUÇÃO [mA(),mC()]

## 7. Análise

O resultado final gerado pelo CAAMPI4J é uma listagem dos grupos identificados contendo o identificador do grupo, a pontuação obtida pela medida de ordenação, os métodos pertencentes a cada grupo e os pontos de corte identificados.

De forma a exemplificar a saída com um sistema real, na Tabela 5.9 são apresentados os quatro primeiros resultados ordenados pela medida GSRank que foram obtidos a partir da execução do CAAMPI4J no sistema JHotDraw, utilizando o algoritmo de agrupamento Hierárquico e a medida de distância  $D_{SOND}$ . Para abreviar, os métodos são exibidos somente com o identificador, nome de classe e nome do método, e os pontos de conte são exibidos somente com os identificadores.

Tabela 5.9: Exemplo Resultado CAAMPI4J

Grupo	GSRank	Métodos	Pontos de Corte
K255	1084	[1574]Figure.displayBox() [1894]AbstractFigure.displayBox() [1707]AnimationDecorator.displayBox() [1152]GroupFigure.displayBox() [2556]StandardDrawing.displayBox() [315]GraphicalCompositeFigure.displayBox() [1056]BorderDecorator.displayBox() [2256]DecoratorFigure.displayBox() [1325]RectangleFigure.displayBox() [1337]RoundRectangleFigure.displayBox() [1364]TextFigure.displayBox() [1786]NodeFigure.displayBox() [804]TextAreaFigure.displayBox() [1110]EllipseFigure.displayBox() [169]ComponentFigure.displayBox() [664]PolygonFigure.displayBox() [1258]PolyLineFigure.displayBox() [1163]ImageFigure.displayBox() [1232]NullFigure.displayBox() [1827]PertFigure.displayBox()	$\emptyset$
K161	970	[1635]FigureEnumeration.nextFigure() [2344]FigureEnumerator.nextFigure() [2501]ReverseFigureEnumerator.nextFigure() [2544]SingleFigureEnumerator.nextFigure() [2322]FigureAndEnumerator.nextFigure() [2323]FigureAndEnumerator.hasNextFigure() [1636]FigureEnumeration.hasNextFigure() [2343]FigureEnumerator.hasNextFigure() [2500]ReverseFigureEnumerator.hasNextFigure() [2543]SingleFigureEnumerator.hasNextFigure()	1635 APOS A CHAMADA 1636 1635 ANTES DA CHAMADA 1571 1636 APOS A CHAMADA 1580

Continua na próxima página ...

... continuação da página anterior			
Grupo	GSRank	Métodos	Pontos de Corte
K147	632	[1178]InsertImageCommand\$UndoActivity.undo() [1428]TextTool\$UndoActivity.undo() [1065]BorderTool\$UndoActivity.undo() [2508]SelectAllCommand\$UndoActivity.undo() [2307]DragTracker\$UndoActivity.undo() [2108]ChangeConnectionHandle\$UndoActivity.undo() [714]PolygonScaleHandle\$UndoActivity.undo() [882]TriangleRotationHandle\$UndoActivity.undo() [1134]FontSizeHandle\$UndoActivity.undo() [1299]PolyLineHandle\$UndoActivity.undo() [1316]RadiusHandle\$UndoActivity.undo() [2036]ResizeHandle\$UndoActivity.undo() [1080]ConnectedTextTool\$UndoActivity.undo() [1085]ConnectedTextTool\$DeleteUndoActivity.undo() [2012]AlignCommand\$UndoActivity.undo() [2070]ChangeAttributeCommand\$UndoActivity.undo() [2244]CutCommand\$UndoActivity.undo() [2291]DeleteCommand\$UndoActivity.undo() [1147]GroupCommand\$UndoActivity.undo() [1441]UngroupCommand\$UndoActivity.undo() [2466]PasteCommand\$UndoActivity.undo() [2534]SendToBackCommand\$UndoActivity.undo() [851]TextAreaTool\$UndoActivity.undo() [2218]ConnectionTool\$UndoActivity.undo() [3048]Undoable.undo() [3060]UndoableAdapter.undo()	3060 ANTES DA EXECUCAO [851, 1134, 2307, 2508, 714, 882, 1065, 2036, 1428, 2070, 1299, 2534, 2466, 2291, 2012, 2108, 1441, 2218, 1147, 2244, 1178, 1316]
K354	463	[3075]UndoableCommand.execute() [2841]Command.execute() [2976]RedoCommand.execute() [3143]UndoCommand.execute() [2315]DuplicateCommand.execute() [2461]PasteCommand.execute() [2066]ChangeAttributeCommand.execute() [2058]BringToFrontCommand.execute() [2530]SendToBackCommand.execute() [1173]InsertImageCommand.execute() [2240]CutCommand.execute() [2287]DeleteCommand.execute() [2504]SelectAllCommand.execute() [2007]AlignCommand.execute() [1143]GroupCommand.execute() [1437]UngroupCommand.execute() [962]ZoomCommand.execute() [1860]AbstractCommand.execute() [2223]CopyCommand.execute() [2674]ToggleGridCommand.execute()	1860 ANTES DA EXECUCAO [2674, 2066, 2007, 2504, 2315, 2976, 2530, 1437, 2287, 2223, 1173, 2461, 1143, 962, 2058, 3143, 2240]

Neste exemplo, o analista inicia pelo grupo de maior pontuação, no caso o K255. O grupo K255 é composto por vários métodos `displayBox`. Analisando o código fonte dos métodos `displayBox`, verifica-se que estes são métodos de acesso a atributos que não foram eliminados na fase de filtro devido ao fato de não possuírem o prefixo “get”. Portanto, não são interesses transversais.

Exemplo no código fonte (`RectangleFigure.java`):

```
private Rectangle fDisplayBox; //Atributo
public Rectangle displayBox() {
    return new Rectangle(
        fDisplayBox.x,
        fDisplayBox.y,
        fDisplayBox.width,
        fDisplayBox.height);
} //Retorna uma nova instância, cópia do atributo
}
```

Analisando o segundo grupo, K161, este se trata de um grupo de métodos utilitários. Estes métodos fazem a simples chamada aos métodos da classe da plataforma Java `Iterator`. Portanto, não são interesses transversais.

Exemplo no código fonte (`FigureEnumerator.java`):

```
import java.util.Iterator; //Iterator pertence a plataforma Java
private Iterator myIterator; //Atributo da classe Iterator
public boolean hasNextFigure() {
    return myIterator.hasNext(); //Invocação a Iterator
}
public Figure nextFigure() {
    return (Figure)myIterator.next(); //Invocação a Iterator
}
```

Em análise ao terceiro grupo, K147, este é um interesse transversal do tipo Undo. Este interesse transversal armazena o estado histórico de um objeto de forma que se possa desfazer as alterações efetuadas. A lógica para capturar, armazenar e desfazer as alterações pode ser encapsulada em aspectos. No JHotDraw, o método 3060 verifica se as atividades executadas sobre o objeto podem ser desfeitas:

```
public class UndoableAdapter implements Undoable {
    /*
     * Undo the activity
     * @return true if the activity could be undone, false otherwise
     */
    public boolean undo() { //Método [3060]UndoableAdapter.undo()
        return isUndoable();
    }
    (...)
}
```

Todos os métodos undo estendem UndoableAdapter e tem como primeira operação a invocação ao método 3060 de forma a verificar o seu respectivo estado.

Exemplo UndoActivity, que estende UndoableAdapter e invoca o método 3060:

```
public static class UndoActivity extends UndoableAdapter
    public boolean undo() {
        if (!super.undo()) { //Invocação a [3060]UndoableAdapter.undo()
            return false;
        }
        (...)
    }
}
```

O ARPIM conseguiu identificar o ponto de corte de 3060 antes da execução de todos os métodos undo. No AJHotDraw o método 3060 foi refatorado como um ponto de corte que captura os comandos que permitem que as atividades sejam desfeitas, substituindo o antigo método.

```
public aspect UndoableCommand{
    pointcut undoableCommands() :
    (
        (target(AlignCommand) && !within(AlignCommand) && !within(AlignCommandUndo))
        || (target(BringToFrontCommand) && !within(BringToFrontCommand) &&
            !within(BringToFrontCommandUndo))
        || (target(ChangeAttributeCommand) && !within(ChangeAttributeCommand) &&
            !within(ChangeAttributeCommandUndo))
        || (target(CutCommand) && !within(CutCommand) && !within(CutCommandUndo))
        || (target>DeleteCommand) && !within>DeleteCommand) && !within>DeleteCommandUndo))
        || (target>DuplicateCommand) && !within>DuplicateCommand) && !within>DuplicateCommandUndo))
        || (target>PasteCommand) && !within>PasteCommand) && !within>PasteCommandUndo))
        || (target>GroupCommand) && !within>GroupCommand) && !within>GroupCommandUndo))
        || (target>InsertImageCommand) && !within>InsertImageCommand)
        || (target>SelectAllCommand) && !within>SelectAllCommand) )
        || (target>SendToBackCommand) && !within>SendToBackCommand))
        || (target>UngroupCommand) && !within>UngroupCommand))
    ) && !within(UndoableCommand);
}
```

Os demais métodos que implementam undo foram encapsulados em aspectos, retirando esta funcionalidade das classes.

O quarto grupo, K354, é um interesse transversal do tipo Command. Este interesse transversal encapsula execução de ações que afetam várias classes e métodos. A lógica para a execução de comandos pode ser encapsulada em aspectos. No JHotDraw o método 1860 é utilizado para checar uma pré-condição da execução de ações, que é a verificação se o editor gráfico esta carregado:



```

public abstract class AbstractCommand implements Command, ... {
    public void execute() {
        if (view() == null) {
            throw new JHotDrawRuntimeException("execute should NOT be getting called
                when view() == null");
        }
    }
    (...)
}

```

Todos os métodos `execute` invocam o método 1860 no início da execução de forma a verificar esta pré-condição. Quando esta pré-condição não é atendida, uma exceção será gerada.

```

public class PasteCommand extends FigureTransferCommand {
    public void execute() {
        super.execute(); //Invocação [1860]AbstractCommand.execute(), Exceção se view==null
    }
    (...)
}

```

O ARPIM conseguiu identificar o ponto de corte de 1860 no início da execução dos métodos das classes `Command`. No `AJHotDraw` o método 1860 foi refatorado através de um ponto de corte que captura o início da execução dos métodos `execute` e sua lógica foi migrada para um adendo:

```

public aspect CommandContracts{
    pointcut commandExecuteCheckView(AbstractCommand acommand) :
        this(acommand)
        && execution(void AbstractCommand+.execute())
        && !within(*..DrawApplication.*)
            && !within(*..CTXWindowMenu.*)
            && !within(*..WindowMenu.*)
            && !within(*..JavaDrawApp.*);

    /**
     * The code checking the reference is crosscutting -
     * it was moved from AbstractCommand.execute() to
     * this advice. The advice implements the pre-condition
     * check in the Command elements.
     */
    before(AbstractCommand acommand) : commandExecuteCheckView(acommand) {
        if (acommand.view() == null) {
            throw new JHotDrawRuntimeException("execute should NOT be getting called
                when view() == null");
        }
    }
    (...)
}

```

Os métodos `execute` foram refatorados, eliminando a checagem de pré-condição:

```

public class PasteCommand extends FigureTransferCommand {
    /**
     * @AJHD refactored: consistent condition check
     * @see CommandContracts
     */
    public void execute() {
        //AJHD refactored
        //super.execute();
    }
    (...)
}

```

Este exemplo ilustra como o analista interage no processo de refatoração com o auxílio do CAAMPI4J. A listagem completa de todas as instâncias de interesses transversais identificadas e validadas nos sistemas `JHotDraw`, `Tomcat` e `HSQLDB`

estão disponíveis no Apêndice A. A listagem completa de todos os grupos identificados na fase de agrupamento com os respectivos métodos e pontuação obtida estão disponíveis no Apêndice C, enquanto os pontos de corte identificados para estes grupos estão disponíveis no Apêndice D.

Além da informação dos grupos, o analista também pode aproveitar outros recursos disponibilizados pela CAAMPI4J, como: 1) as regras que foram extraídas na fase 2 do ARPIM (Formato SE  $a \Rightarrow b$ ); 2) a listagem individual dos métodos ordenada pelo *Fan-In* (Apêndice C); e 3) a navegação pela relação de invocações entre os métodos extraída na fase de computação com o intuito de buscar sementes de interesses transversais (Navegador Dedicado).

## 5.6 Trabalhos Relacionados

A abordagem CAAMPI foi baseada em vários estudos existentes na literatura de Mineração de Aspectos. A seguir são citados todos os trabalhos relacionados:

- Processo CAAMPI: O processo CAAMPI teve como embasamento os processos e fases coletados em vários estudos encontrados na literatura. As fases de Computação, Filtro de Métodos, Agrupamento e Análise Manual foram adaptadas dos estudos [57], [56], [12] e [13]. As fases de Cálculo do *Fan-In* e Ordenação de Grupos foram baseadas no estudo [72]. A fase de Identificação de Pontos de Corte teve como base o estudo [27].
- Medida de Distância  $D_S$ : A medida de distância de espalhamento é uma adaptação das medidas de distância utilizadas nos estudos [13] e [12]. As principais diferenças da medida proposta são que o intervalo de saída foi limitado entre 0 e 1 de forma a tornar a medida compatível com mais algoritmos de agrupamento e a inclusão do polimorfismo na coleção de entrada  $Col_S(m)$ . A inclusão do polimorfismo foi baseada no estudo [43].
- Medida de Distância  $D_O$ : A medida de distância de operações foi idealizada com base na técnica de detecção de clones em árvores de sintaxe abstrata proposta no estudo [8]. Enquanto na detecção de clones são identificados quaisquer fragmentos de código duplicado, na medida de operações são contabilizadas somente as invocações efetuadas a métodos, e destas, somente as invocações efetuadas a métodos remanescentes após a fase de filtro de métodos.
- Medida de Distância  $D_N$ : A medida de distância de nomes foi idealizada com base nas medidas de distância propostas nos estudos [62] e [72]. Em [62] é utilizada uma medida de distância baseada no tamanho da maior sub cadeia de caracteres comum entre os nomes dos métodos. Já no estudo [72] é utilizada uma medida de distância que quebra o nome dos métodos em palavras e compara a quantidade de palavras idênticas no nome dos métodos. Em contraste, a medida de distância de nomes faz o uso da distância de edição de Levenshtein [40], que mede a quantidade de inclusões, substituições ou exclusões de letras que são necessárias para transformar um nome no outro. Além disto, a distância de nomes não avalia somente o nome do método como em [62] e [72], mas também a distância de edição do nome da classe a qual o método pertence, a distância do conjunto de tipos de parâmetros e se o tipo do retorno é idêntico ou não.

- Medidas de Ordenação de grupos: A medida de ordenação de grupos GFRank teve como embasamento o método para ordenação de grupos utilizado por Zhang et al no estudo [72]. A medida foi formalizada e contextualizada, porém, não houve alteração em relação a ideia original dos autores. Já a interconectividade interna utilizada pelas medidas GSIRank e GFIRank foi baseada no cálculo da Interconectividade Relativa presente no estudo [33].
- Método de Identificação de pontos de corte ARPIM: O método de identificação de pontos de corte ARPIM teve como base o processo de extração de regras utilizado pelos autores He et al no trabalho [27]. No trabalho, He et al finalizam o processo de extração de regras logo após a fase de filtro de regras, sendo que não são extraídas regras de antecedência e nem é gerado vínculo das regras com os grupos de candidatos a aspectos. He et al também definiram dois critérios simples para serem aplicados diretamente às regras obtidas:
  - Ponto de Corte de Execução: Quando a pré-condição ou pós-condição de regras for *inicio()* ou *fim()*.
  - Ponto de Corte de Chamada: Quando a pré-condição e pós-condição não contiverem os itens *inicio()* ou *fim()*.

O ARPIM difere do estudo [27] devido aos seguintes aperfeiçoamentos: Fase de filtro captura tanto regras de sequência quanto as regras de antecedência, inclusão da fase de classificação de regras para separar as regras por tipo, definição de critérios específicos para a identificação de cada um dos tipos de ponto de corte, vinculação dos resultados de candidatos a pontos de corte com o resultado de candidatos a aspectos.

- Algoritmo de seleção de métodos representativos: O algoritmo de seleção de métodos representativos (Algoritmo 3) é uma adaptação do algoritmo de seleção dos grupos iniciais utilizado nos estudos [57] [12] [56] e [13]. A principal diferença é que o algoritmo proposto foi adaptado para se tornar independente do algoritmo de agrupamento. Portanto, a versão proposta pode ser utilizada tanto para recuperar os métodos representativos quanto a quantidade de métodos representativos. A lógica principal de identificação dos métodos representativos foi preservada.

## 5.7 Considerações Finais

Neste capítulo foi apresentada a abordagem CAAMPI. A CAAMPI introduz um processo completo que contempla também a ordenação de grupos e a identificação de candidatos a pontos de cortes. As fases adotadas na CAAMPI são fruto da pesquisa das melhores características encontradas nos estudos existentes na literatura de mineração de aspectos.

Além disto, foram definidas as medidas de distância  $D_{SOND}$ ,  $D_S$ ,  $D_O$  e  $D_N$ , que são fruto de pesquisa dos vários tipos de sintomas que os interesses transversais podem assumir em um sistema, e, as medidas de ordenação GSRank, GSIRank, GFRank e GFIRank, que geram uma pontuação dos grupos, indicando ao analista quais são os grupos que devem ser priorizados na análise.

A abordagem CAAMPI também contribui com a definição do ARPIM, um método completo de identificação de pontos de corte que se interliga com a identificação de candidatos a aspectos. O método auxilia os analistas tanto na análise, subsidiando com

informações de como os métodos de um determinado candidato a aspecto se comportam no sistema, bem como no planejamento da refatoração, subsidiando com informações de quais pontos de cortes podem existir no sistema refatorado.

Já o *framework* CAAMPI4J contribui com a definição técnica de implementação da abordagem CAAMPI em linguagem Java, sendo que o CAAMPI4J também pode ser aproveitado como um guia para a criação de *frameworks* de implementação da abordagem CAAMPI para outras linguagens.

De forma a comprovar na prática a eficácia da abordagem CAAMPI, o próximo capítulo é destinado a execução de experimentos práticos, com a coleta e análise dos resultados. A CAAMPI foi avaliada em três perspectivas diferentes: eficácia na identificação de candidatos a aspectos, eficácia na ordenação de grupos e eficácia na identificação de pontos de corte.

## CAPÍTULO 6

### AValiação DA ABORDAGEM CAAMPI

Neste capítulo é avaliada a eficácia da abordagem CAAMPI no agrupamento de candidatos a aspectos, ordenação de grupos e identificação de pontos de corte.

Para a avaliação foram utilizados três sistemas reais e índices de avaliação quantitativos em experimentos que exploram todas as possibilidades de combinações dos algoritmos e medidas de distância implementados pelo CAAMPI4J.

Este capítulo descreve a coleta e análise dos resultados obtidos e está organizado nas seguintes seções: Na Seção 6.1 é descrita a metodologia utilizada para efetuar os experimentos, nas Seções 6.2, 6.3 e 6.4 são apresentados os resultados relativos respectivamente as fases de agrupamento, ordenação de grupos e identificação de pontos de corte.

#### 6.1 Metodologia

Nesta seção é apresentada a metodologia que foi utilizada nos experimentos. São componentes da metodologia: o seu objetivo, a descrição dos sistemas utilizados como estudos de caso, os índices de avaliação de qualidade utilizados e os passos seguidos para a execução dos experimentos.

##### 6.1.1 Objetivo dos Experimentos

O objetivo principal dos experimentos é a avaliação da eficácia da abordagem CAAMPI na mineração de aspectos, ordenação de grupos e identificação de pontos de corte.

A avaliação da fase de mineração de aspectos consiste na comparação de três algoritmos de agrupamento combinados com seis medidas de distância. A abordagem é avaliada quanto à qualidade dos grupos de candidatos a aspectos obtidos.

Quanto à ordenação de grupos, a avaliação consiste na comparação de quatro medidas de ordenação. A abordagem é avaliada quanto à qualidade da classificação dos grupos de forma a reduzir o universo de grupos exigidos para análise manual.

Em relação a fase de identificação de pontos de corte, a avaliação consiste na determinação do nível de cobertura dos pontos de corte, e, no aproveitamento dos pontos de corte identificados na refatoração do sistema para aspectos.

##### 6.1.2 Sistemas utilizados

Nesta seção são apresentados os três sistemas que foram utilizados.

###### JHotDraw

O JHotDraw [31] é um editor gráfico para a manipulação de figuras através de uma interface gráfica com o usuário. Nos experimentos foi adotada a versão 5.4b1 devido ao fato da existência de estudos sobre os interesses transversais constantes nesta versão e especialmente por ser a base da versão refatorada para aspectos AJHotDraw [2].

### Apache Tomcat

O Apache Tomcat [6] é um servidor WEB utilizado para hospedagem de páginas HTML e aplicativos Java escritos nas tecnologias *Servlet* e *JavaServer Pages* (JSP). Nos experimentos foi adotada a versão 5.5.17 devido ao fato da existência de estudos sobre os interesses transversais constantes nesta versão.

### HSQldb

O HSQldb [29] é um servidor de banco de dados relacional que possui como destaque as funcionalidades de suporte total a linguagem SQL ANSI, gerenciamento de transações, persistência em disco ou em memória e execução em modo servidor ou embutido como biblioteca em outras aplicações. Nos experimentos foi adotada a versão 1.8.0.2 devido ao fato desta ser a base da versão refatorada para aspectos AJHSQldb [3].

## 6.1.3 Índices de avaliação de qualidade de agrupamento

De forma que seja possível avaliar e comparar os algoritmos de agrupamento e medidas de distância quanto a mineração de aspectos, se torna necessária a definição de índices de avaliação de qualidade padronizados para mensurar a qualidade do agrupamento obtido por cada um dos algoritmos. Além disto, é necessária uma lista contendo todas as instâncias de interesses transversais validadas por um especialista.

O levantamento da lista de instâncias de interesses transversais e a sua respectiva validação foi efetuada através dos seguintes recursos: 1) Resultados da mineração de aspectos dos sistemas JHotDraw e Tomcat com análise de *fan-in* reportados em [41] e analisados em [43]; 2) Estudo de refatoração para aspectos do JHotDraw disponível em [42]; 3) Grupos de candidatos a aspectos com as maiores pontuações na medida de ordenação GSIRank; e 4) Conciliação das versões originais com as versões orientadas a aspectos do JHotDraw e HSQldb. A lista de interesses transversais validada pode ser consultada no Apêndice A, e, especificamente para os sistemas JHotDraw e HSQldb, que possuem versões orientadas a aspectos, também podem ser consultadas as evidências contendo o mapeamento das instâncias validadas com os respectivos aspectos refatorados.

Para a definição dos índices, foi considerado o princípio de que uma partição ótima é aquela em que cada grupo representa somente um interesse transversal, e, cada interesse transversal é representado por apenas um grupo. Portanto, foram adotados dois índices de avaliação de qualidade complementares: a dispersão, para verificar se os métodos pertencentes a um interesse transversal estão no mesmo grupo, e, a diversificação, para verificar se os métodos pertencentes a um grupo são referentes ao mesmo interesse transversal.

Os índices adotados são descritos a seguir.

### Índice de qualidade de dispersão

Este índice foi proposto por Grigoreta et al no estudo [21] e define a qualidade do agrupamento quanto ao grau de dispersão das instâncias de interesses transversais em grupos de candidatos a aspectos. Este índice segue o propósito de que cada instância de interesse transversal deve ser classificada pelo algoritmo de agrupamento em um único grupo candidato a aspecto. Se uma instância de interesse transversal estiver presente em mais de um grupo candidato a aspecto, o interesse transversal foi identificado com dispersão em múltiplos grupos candidatos a aspectos.

O cálculo do índice foi efetuado conforme o índice DISP descrito na Seção 4.5.

A saída deste índice é a média da dispersão das instâncias dos interesses transversais em grupos e é compreendida no intervalo entre 0 e 1. Quanto maior o valor, melhor o índice de dispersão geral dos interesses transversais. A saída 1 indica o valor ideal, na qual nenhuma das instâncias de interesses transversais apresentou dispersão.

### Índice de qualidade de diversificação

Este índice foi proposto inicialmente por Grigoreta et al no estudo [21] e define a qualidade do agrupamento quanto ao grau de diversificação dos grupos de candidatos a aspectos encontrados. Este índice segue o propósito de que cada grupo encontrado pelo algoritmo de agrupamento deve possuir métodos pertencentes a uma única instância de interesse transversal, ou, se tratar completamente de interesses base. Se o grupo possui heterogeneidade de interesses transversais, ou, possui interesses base junto a interesses transversais, o grupo possui diversidade.

O cálculo do índice foi adaptado do índice DIV descrito na Seção 4.5. O índice DIV computa a média de diversificação de todos os grupos, inclusive aqueles que contém somente interesses base. Esta característica interfere no resultado da avaliação, pois os grupos que contém somente interesses base são considerados ótimos. Este fato beneficia indevidamente os algoritmos de agrupamento que geram mais grupos, já que, quanto maior a quantidade de grupos de interesses base, mais grupos sem dispersão existirão, e portanto, melhor será o índice obtido. De forma a eliminar esta limitação do índice original, o índice de DIV foi adaptado para desconsiderar os grupos compostos somente por interesses base, e portanto, obter o mesmo resultado independentemente da quantidade de grupos de interesses base gerados pelo algoritmo de agrupamento. A alteração consiste na substituição da Equação 4.19 da Seção 4.5 pela Equação 6.1:

$$diversificacao(IT, \mathcal{K}) = \frac{1}{n} \sum_{i=1}^n \{div(IT, \mathcal{K}_i) | \mathcal{K}_i \cap IT \neq \emptyset\} \quad (6.1)$$

Onde:  $n$  (Equação 6.2) é a quantidade de grupos de candidatos a aspectos do conjunto  $\mathcal{K}$  que contém ao menos um método pertencente ao conjunto  $IT$ ,  $\mathcal{K}_i$  é um grupo específico pertencente ao conjunto  $\mathcal{K}$  que contém ao menos um método pertencente ao conjunto  $IT$  e  $div(IT, \mathcal{K}_i)$  é a diversidade de um grupo  $\mathcal{K}_i \in \mathcal{K}$ , definida na Equação 4.20 da Seção 4.5.

$$n = \left| \sum_{j=1}^{|\mathcal{K}|} \{\mathcal{K}_j | \mathcal{K}_j \in \mathcal{K} \text{ e } \mathcal{K}_j \cap IT \neq \emptyset\} \right| \quad (6.2)$$

A saída deste índice é a média da diversidade dos grupos encontrados pelo algoritmo de agrupamento e é compreendida no intervalo entre 0 e 1. Quanto maior o valor, melhor o índice de diversidade geral dos grupos. A saída 1 indica o valor ideal, na qual nenhum dos grupos apresentou diversidade.

### 6.1.4 Índices de avaliação de qualidade de ordenação de grupos

De forma que seja possível avaliar e comparar as medidas de ordenação de grupos, se torna necessária a definição de índices de avaliação de qualidade padronizados para mensurar a

qualidade da ordenação obtida por cada uma das medidas.

Partindo do princípio de que uma ordenação ótima é aquela em que os  $n$  primeiros grupos com as maiores pontuações obtidas pertencem ao conjunto de  $n$  instâncias de interesses transversais, foram criados três índices de avaliação: liderança, economia e colocação.

Os índices criados são detalhados a seguir.

### Índice de qualidade de liderança

O objetivo deste índice é definir a qualidade da ordenação quanto aos grupos que estão na liderança, ou seja, os grupos que obtiveram as primeiras colocações quanto a pontuação. Este índice segue o propósito de que os  $n$  primeiros colocados devem pertencer ao conjunto interesses transversais. Se um ou mais grupos entre os  $n$  colocados não pertencerem ao conjunto de interesses transversais, esta é uma indicação de que a medida priorizou grupos que não deveriam estar na liderança de pontuação.

Sendo  $\mathcal{K}$  o conjunto de candidatos a aspectos encontrados pelo algoritmo de agrupamento,  $\mathcal{L}$  um subconjunto de  $\mathcal{K}$  com os grupos de candidatos a aspectos que obtiveram as  $n$  primeiras colocações na medida de ordenação de grupos,  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista, e  $n$  a quantidade de grupos de candidatos a aspectos em que ao menos um método pertence ao conjunto de instâncias de interesses transversais  $IT$ . O índice de qualidade de liderança  $lideranca(\mathcal{K}, \mathcal{L}, IT)$  de um subconjunto de candidatos a aspectos líderes  $\mathcal{L}$  em respeito a um conjunto de instâncias de interesses transversais  $IT$  é definido na Equação 6.3:

$$lideranca(\mathcal{K}, \mathcal{L}, IT) = \frac{|Col_{lid}(\mathcal{L}, IT)|}{n} \quad (6.3)$$

Onde:  $n$  (Equação 6.2) é a quantidade de grupos de candidatos a aspectos do conjunto  $\mathcal{K}$  que contêm ao menos um método pertencente ao conjunto  $IT$  e  $Col_{lid}(\mathcal{L}, IT)$  é o conjunto de grupos do subconjunto  $\mathcal{L}$  em que ao menos um método pertence ao conjunto de interesses transversais  $IT$ , ou seja:

$$Col_{lid}(\mathcal{L}, IT) = \sum_{i=1}^n \{\mathcal{L}_i | \mathcal{L}_i \in \mathcal{L} \text{ e } \mathcal{L}_i \cap IT \neq \emptyset\} \quad (6.4)$$

A saída deste índice é a qualidade de liderança do conjunto e é compreendida no intervalo entre 0 e 1. Quanto maior o valor, melhor o índice de liderança. A saída 1 indica o valor ideal, na qual todos os elementos do conjunto de liderança pertencem ao conjunto de interesses transversais.

### Índice de qualidade de economia

O objetivo deste índice é definir a qualidade da ordenação quanto a economia do esforço de análise que deve ser efetuado pelo analista, ou seja, o quanto de esforço do analista que foi reduzido com a aplicação da medida. Este índice segue o propósito de que todos os  $n$  primeiros grupos devem pertencer ao conjunto de interesses transversais, sendo que neste estado a economia do esforço é total. Se o último colocado não estiver na  $n$ -ésima posição, esta é uma indicação de que a medida não foi totalmente eficaz na economia do esforço de análise.



Sendo  $\mathcal{K}$  o conjunto de candidatos a aspectos encontrados pelo algoritmo de agrupamento e ordenados pela medida de ordenação,  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista. O índice de qualidade de economia  $economia(\mathcal{K}, IT)$  de um conjunto de candidatos a aspectos ordenados  $\mathcal{K}$  em respeito a um conjunto de instâncias de interesses transversais  $IT$  é definido na equação 6.5:

$$economia(\mathcal{K}, IT) = 1 - \frac{ultimo - n}{|\mathcal{K}| - n} \quad (6.5)$$

Onde: *ultimo* é a posição, ordinal, da última colocação obtida por um grupo candidato a aspecto em que ao menos um método pertence ao conjunto de interesses transversais  $IT$  e  $n$  (Equação 6.2) é a quantidade de grupos de candidatos a aspectos do conjunto  $\mathcal{K}$  que contêm ao menos um método pertencente ao conjunto  $IT$ .

A saída deste índice é a qualidade da ordenação quanto a economia do esforço de análise e é compreendida no intervalo entre 0 e 1. Quanto maior o valor, melhor o índice de economia. A saída 1 indica o valor ideal, na qual o último elemento está na  $n$ -ésima posição e, portanto, não houve esforço na análise de grupos que não são interesses transversais.

### Índice de qualidade de colocação

O objetivo deste índice é definir a qualidade da ordenação quanto a colocação dos grupos de candidatos a aspectos que pertencem ao conjunto de interesses transversais, ou seja, o quanto a medida foi eficaz em classificar os grupos de interesses transversais entre as primeiras colocações. Este índice segue o propósito de que os grupos que possuem interesses transversais devem estar entre as  $n$  primeiras colocações. Se um ou mais grupos estiverem além das  $n$  primeiras colocações, esta é uma indicação de que a medida não foi eficaz quanto a colocação dos grupos de candidatos a aspectos que pertencem ao conjunto de interesses transversais.

Sendo  $\mathcal{K}$  o conjunto de candidatos a aspectos encontrados pelo algoritmo de agrupamento e ordenados pela medida de ordenação,  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista. O índice de qualidade de colocação  $colocacao(\mathcal{K}, IT)$  de um conjunto de candidatos a aspectos ordenados  $\mathcal{K}$  em respeito a um conjunto de instâncias de interesses transversais  $IT$  é definido na Equação 6.6:

$$colocacao(\mathcal{K}, IT) = 1 - \frac{obtido(\mathcal{K}, IT) - otimo(\mathcal{K}, IT)}{pior(\mathcal{K}, IT) - otimo(\mathcal{K}, IT)} \quad (6.6)$$

Onde:  $otimo(\mathcal{K}, IT)$  é a soma das colocações do caso ótimo, ou seja, em que os grupos que contêm interesses transversais foram classificados nas  $n$  primeiras posições, calculado conforme a Equação 6.7:

$$otimo(\mathcal{K}, IT) = \frac{(1 + n) * n}{2} \quad (6.7)$$

Onde  $n$  (Equação 6.2) é o conjunto de grupos de candidatos a aspectos do conjunto  $\mathcal{K}$  que contêm ao menos um método pertencente ao conjunto  $IT$  e  $pior(\mathcal{K}, IT)$  é a

soma das colocações do pior caso, ou seja, em que os grupos que contêm interesses transversais foram classificados nas  $|\mathcal{K}|$  últimas posições, calculado conforme a Equação 6.8:

$$pior(\mathcal{K}, IT) = \frac{((|\mathcal{K}| - n) + |\mathcal{K}|) * n}{2} \quad (6.8)$$

E,  $obtido(\mathcal{K}, IT)$  é a soma das colocações dos grupos de  $\mathcal{K}$  que contêm interesses transversais, calculado conforme a Equação 6.9:

$$obtido(\mathcal{K}, IT) = \sum_{i=1}^{|\mathcal{K}|} \{i | \mathcal{K}_i \in \mathcal{K} \text{ e } \mathcal{K}_i \cap IT \neq \emptyset\} \quad (6.9)$$

A saída deste índice é a qualidade da ordenação quanto a colocação dos grupos de candidatos a aspectos que pertencem ao conjunto de interesses transversais e é compreendida no intervalo entre 0 e 1. Quanto maior o valor, melhor o índice de colocação. A saída 1 indica o valor ideal, na qual os grupos de candidatos a aspectos que pertencem ao conjunto de interesses transversais ocuparam as  $n$  primeiras posições.

### 6.1.5 Índices de avaliação de qualidade de identificação de pontos de corte

De forma que seja possível avaliar o método de identificação de pontos de corte, torna-se necessária a definição de índices de avaliação de qualidade padronizados para mensurar a qualidade dos pontos de corte gerados.

Partindo do princípio de que uma identificação de pontos de corte ótima é aquela em que para todo grupo de candidatos a aspectos que possuam instâncias de interesses transversais, haja ao menos um ponto de corte identificado, e, dos pontos de corte identificados, os mesmos sejam realmente aplicáveis e aproveitados na refatoração do sistema, foram criados dois índices de avaliação: cobertura e aproveitamento.

Os índices criados são descritos com detalhes a seguir.

#### Índice de qualidade de cobertura

O objetivo deste índice é definir a qualidade da identificação de pontos de corte quanto a cobertura de pontos de corte aos grupos que contêm métodos pertencentes ao conjunto de interesses transversais. Este índice segue o propósito de que todos os grupos que contêm métodos pertencentes ao conjunto de interesses transversais devem possuir ao menos um ponto de corte. Se um ou mais grupos não possuírem ponto de corte identificado, esta é uma indicação de que o método não obteve cobertura completa.

Sendo  $\mathcal{K}$  o conjunto de candidatos a aspectos encontrados pelo algoritmo de agrupamento,  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista, e  $PC$  o conjunto de pontos de corte identificados pelo método de identificação de pontos de corte referentes a interesses transversais. O índice de qualidade de cobertura  $cobertura(PC, \mathcal{K}, IT)$  de um conjunto de pontos

de corte  $PC$  em respeito a um grupo de candidatos a aspectos  $\mathcal{K}$  e um conjunto de interesses transversais  $IT$  é definido na Equação 6.10:

$$cobertura(PC, \mathcal{K}, IT) = \frac{|Col_{cob}(PC, \mathcal{K}, IT)|}{n} \quad (6.10)$$

Onde:  $n$  (Equação 6.2) é a quantidade de grupos de candidatos a aspectos do conjunto  $\mathcal{K}$  que contêm ao menos um método pertencente ao conjunto  $IT$  e  $Col_{cob}(PC, \mathcal{K}, IT)$  é o conjunto de grupos do conjunto  $\mathcal{K}$  em que ao menos um método pertence ao conjunto de interesses transversais  $IT$ , e, que ao menos um método possui ponto de corte no conjunto  $PC$ , ou seja:

$$Col_{cob}(PC, \mathcal{K}, IT) = \sum_{i=1}^{|\mathcal{K}|} \{\mathcal{K}_i | \mathcal{K}_i \in \mathcal{K} \text{ e } \mathcal{K}_i \cap IT \neq \emptyset \text{ e } \mathcal{K}_i \cap PC \neq \emptyset\} \quad (6.11)$$

A saída deste índice é a qualidade do método de identificação de pontos de corte quanto a cobertura e é compreendida no intervalo entre 0 e 1. Quanto maior o valor, melhor o índice de cobertura. A saída 1 indica o valor ideal, na qual todos os grupos que contêm métodos referentes a interesses transversais foram cobertos por ao menos um ponto de corte.

### Índice de qualidade de aproveitamento

O objetivo deste índice é definir a qualidade da identificação de pontos de corte quanto ao aproveitamento dos pontos de corte identificados na refatoração para aspectos. Este índice segue o propósito de que todos os pontos de corte referentes a métodos pertencentes a instâncias de interesses transversais devem ter sido refatorados na versão orientada a aspectos do sistema avaliado. Se um ou mais pontos de corte não foram refatorados, esta é um indicação que os pontos de corte identificados não são totalmente aproveitados.

Sendo  $PC$  o conjunto de pontos de corte identificados pelo método de identificação de pontos de corte referentes a interesses transversais e  $IT$  o conjunto de instâncias de interesses transversais identificadas e validadas por um especialista. O índice de qualidade de aproveitamento  $aproveitamento(PC, IT)$  de um conjunto de pontos de corte  $PC$  em respeito a um conjunto de instâncias de interesses transversais  $IT$  é definido na Equação 6.12:

$$aproveitamento(PC, IT) = \frac{|Col_{apr}(PC)|}{|PC|} \quad (6.12)$$

Onde:  $Col_{apr}(PC)$  é o conjunto de pontos de corte de  $PC$  que foram aproveitados na versão orientada a aspectos do sistema em análise, ou seja:

$$Col_{apr}(PC) = \sum_{i=1}^{|PC|} \{PC_i | PC_i \in PC \text{ e } PC_i \cap IT \neq \emptyset \text{ e } PC_i \rightarrow \text{Aproveitado}\} \quad (6.13)$$

A saída deste índice é a qualidade do método de identificação de pontos de corte quanto ao aproveitamento e é compreendida no intervalo entre 0 e 1. Quanto maior

o valor, melhor o índice de aproveitamento. A saída 1 indica o valor ideal, na qual todos os pontos de corte referentes a interesses transversais foram aproveitados na versão orientada a aspectos.

### 6.1.6 Etapas dos experimentos

A avaliação da abordagem CAAMPI foi efetuada em três etapas. Na primeira, foi avaliada a qualidade dos grupos de candidatos a aspectos obtidos na fase de agrupamento, na segunda foi avaliada a qualidade na ordenação de grupos e na terceira foi avaliada a qualidade da identificação de pontos de corte.

A metodologia utilizada em cada uma das fases é descrita a seguir.

#### Fase de Agrupamento

Os experimentos para a avaliação da abordagem CAAMPI quanto à mineração de candidatos a aspectos foram efetuados através da execução do CAAMPI4J com vários tipos de algoritmos e medidas de distância. Para cada combinação de algoritmo de agrupamento e medida de distância é dado o nome de instância. Foram gerados ao todo 18 instâncias combinando os algoritmos de agrupamento k-medoids, hierárquico clássico e CHAMELEON e as medidas de distâncias  $D_S^{CCC}$ ,  $D^{HBZH}$ ,  $D_S$ ,  $D_O$ ,  $D_N$  e  $D_{SOND}$ , conforme apresentado na Tabela 6.1.

Tabela 6.1: Instâncias avaliadas

Instância	Agrupamento	Medida
1	k-medoids	$D_S^{CCC}$
2	k-medoids	$D^{HBZH}$
3	k-medoids	$D_S$
4	k-medoids	$D_O$
5	k-medoids	$D_N$
6	k-medoids	$D_{SOND}$
7	Hierárquico clássico	$D_S^{CCC}$
8	Hierárquico clássico	$D^{HBZH}$
9	Hierárquico clássico	$D_S$
10	Hierárquico clássico	$D_O$
11	Hierárquico clássico	$D_N$
12	Hierárquico clássico	$D_{SOND}$
13	CHAMELEON	$D_S^{CCC}$
14	CHAMELEON	$D^{HBZH}$
15	CHAMELEON	$D_S$
16	CHAMELEON	$D_O$
17	CHAMELEON	$D_N$
18	CHAMELEON	$D_{SOND}$

Cada instância gerada para a fase de agrupamento teve seus parâmetros calibrados com cada um dos sistemas. Os parâmetros calibrados são a junção dos parâmetros do algoritmo de agrupamento e os parâmetros da medida de distância.

Os parâmetros calibrados para os algoritmos de agrupamento são:

- *k-medoids* e hierárquico clássico: Os algoritmos de agrupamento *k-medoids* e hierárquico clássico implementados no CAAMPI4J possuem indiretamente o parâmetro *distMin* para calibração. O parâmetro *distMin* é utilizado pelo algoritmo de seleção de métodos representativos, apresentado no Algoritmo 3 e descrito na Seção 5.5. Sua função é criar um limiar quanto a distância mínima que um método pode ter dos demais para ser considerado um método representativo. Nos trabalhos [13], [21] e [12] foram utilizados valores entre 0.75 a 1.00. Testes empíricos demonstraram que valores abaixo de 0.70 e acima de 1.00 não geram bons resultados. Portanto, neste trabalho são consideradas somente configurações no intervalo de 0.70 a 1.00.
- CHAMELEON: O algoritmo de agrupamento hierárquico CHAMELEON implementado no CAAMPI4J possui dois parâmetros:
  - *tam\_part*: Parâmetro utilizado na segunda fase do algoritmo para limitar o tamanho das partições geradas. O algoritmo irá quebrar as partições até que nenhuma partição possua quantidade de elementos maior do que o parâmetro *tam\_part*. Segundo o estudo [33] este parâmetro deve ser menor do que o tamanho da maioria dos grupos que deverão ser gerados, porém, deve ser o suficiente para gerar pequenos grupos. Testes empíricos demonstram que os melhores resultados são obtidos no intervalo entre 3 e 5.
  - *sim\_min*: Parâmetro utilizado na terceira fase do algoritmo como critério de parada para a junção de grupos. O algoritmo irá juntar grupos até que nenhum par de grupos possua similaridade maior que o parâmetro *sim\_min*. Devido ao fato de que em testes empíricos não foram obtidos bons resultados com valores abaixo de 0.8 e acima de 1.2, neste trabalho são testadas configurações no intervalo de 0.8 a 1.2.

Os parâmetros calibrados para as medidas de distância são apresentados na Tabela 6.2. Foram considerados somente os intervalos onde foram obtidos os melhores resultados em testes empíricos.

Tabela 6.2: Parâmetros Calibrados

Medida	Parâmetro	Descrição	Intervalo
$D_N$	$F_{NM}$	Fator de importância do nome do método	[0.40, 1.00]
	$F_{NC}$	Fator de importância do nome da classe	[0.00, 0.40]
	$F_{PR}$	Fator de importância de parâmetros e retorno	[0.00, 0.20]
$D_{SOND}$	$F_S$	Fator de importância da distância de espalhamento	[0.35, 0.50]
	$F_O$	Fator de importância da distância de operações	[0.30, 0.45]
	$F_N$	Fator de importância da distância de nomes	[0.10, 0.30]
	$F_{NM}$	Fator de importância do nome do método	[0.40, 1.00]
	$F_{NC}$	Fator de importância do nome da classe	[0.00, 0.40]
	$F_{PR}$	Fator de importância de parâmetros e retorno	[0.00, 0.20]

Os parâmetros dos algoritmos e das medidas de distância foram combinados e executados nos três sistemas. A melhor configuração escolhida foi aquela que obteve

simultaneamente o maior índice de qualidade de dispersão e diversificação no sistema avaliado. Caso para um determinado sistema nenhuma configuração tenha obtido simultaneamente o maior valor nos dois índices, o critério adotado é a seleção da configuração que obteve a menor diferença de valor para o resultado ótimo (1.00000000), conforme a Equação 6.14.

$$diferenca = (1 - dispersao) + (1 - diversificacao) \quad (6.14)$$

O valor da diferença de índice para o resultado ótimo é compreendido entre 0 e 2, sendo 0 a obtenção simultânea do resultado ótimo nos índices de qualidade de dispersão e diversificação. Sendo assim, quanto mais próximo a 0, melhor é o resultado obtido.

O resultado de cada combinação de configuração de parâmetros pode ser consultado no Apêndice B. A melhor combinação de parâmetros obtida pelas instâncias nos três sistemas é consolidada na Tabela 6.3. Na tabela são apresentados todos os parâmetros possíveis, sendo N/A a indicação de que o parâmetro é não aplicável a instância.

Tabela 6.3: Melhor configuração de parâmetros das instâncias

Instância	Sistema	<i>distMin</i>	<i>tam_part</i>	<i>sim_min</i>	<i>F<sub>S</sub></i>	<i>F<sub>O</sub></i>	<i>F<sub>N</sub></i>	<i>F<sub>NM</sub></i>	<i>F<sub>NC</sub></i>	<i>F<sub>PR</sub></i>
1	JHotDraw	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	HSQLDB	0.80	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	JHotDraw	0.90	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	HSQLDB	0.90	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3	JHotDraw	0.80	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	HSQLDB	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4	JHotDraw	1.00	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	HSQLDB	0.90	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5	JHotDraw	0.80	N/A	N/A	N/A	N/A	N/A	1.00	0.00	0.00
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	1.00	0.00	0.00
	HSQLDB	0.70	N/A	N/A	N/A	N/A	N/A	1.00	0.00	0.00
6	JHotDraw	0.70	N/A	N/A	0.45	0.45	0.10	1.00	0.00	0.00
	Tomcat	0.70	N/A	N/A	0.40	0.35	0.25	0.60	0.20	0.20
	HSQLDB	0.80	N/A	N/A	0.40	0.40	0.20	0.50	0.30	0.20
7	JHotDraw	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	HSQLDB	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
8	JHotDraw	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	HSQLDB	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9	JHotDraw	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	HSQLDB	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10	JHotDraw	1.00	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	JHotDraw	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	JHotDraw	0.70	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
11	JHotDraw	0.70	N/A	N/A	N/A	N/A	N/A	0.50	0.30	0.20
	Tomcat	0.70	N/A	N/A	N/A	N/A	N/A	0.50	0.30	0.20
	HSQLDB	0.70	N/A	N/A	N/A	N/A	N/A	1.00	0.00	0.00
12	JHotDraw	0.70	N/A	N/A	0.40	0.40	0.20	0.50	0.30	0.20
	Tomcat	0.70	N/A	N/A	0.50	0.30	0.20	0.50	0.30	0.20
	HSQLDB	0.70	N/A	N/A	0.50	0.30	0.20	0.40	0.40	0.20
13	JHotDraw	N/A	3	0.8	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	N/A	3	0.8	N/A	N/A	N/A	N/A	N/A	N/A
	HSQLDB	N/A	3	1.0	N/A	N/A	N/A	N/A	N/A	N/A

Continua na próxima página ...

... continuação da página anterior										
Instância	Sistema	<i>distMin</i>	<i>tam_part</i>	<i>sim_min</i>	<i>F<sub>S</sub></i>	<i>F<sub>O</sub></i>	<i>F<sub>N</sub></i>	<i>F<sub>NM</sub></i>	<i>F<sub>NC</sub></i>	<i>F<sub>PR</sub></i>
14	JHotDraw	N/A	3	0.8	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	N/A	3	0.8	N/A	N/A	N/A	N/A	N/A	N/A
	HSQldb	N/A	3	1.2	N/A	N/A	N/A	N/A	N/A	N/A
15	JHotDraw	N/A	4	1.0	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	N/A	3	1.0	N/A	N/A	N/A	N/A	N/A	N/A
	HSQldb	N/A	3	1.0	N/A	N/A	N/A	N/A	N/A	N/A
16	JHotDraw	N/A	3	1.2	N/A	N/A	N/A	N/A	N/A	N/A
	Tomcat	N/A	4	1.2	N/A	N/A	N/A	N/A	N/A	N/A
	HSQldb	N/A	3	1.2	N/A	N/A	N/A	N/A	N/A	N/A
17	JHotDraw	N/A	3	1.2	N/A	N/A	N/A	0.60	0.20	0.20
	Tomcat	N/A	4	1.2	N/A	N/A	N/A	1.00	0.00	0.00
	HSQldb	N/A	4	1.2	N/A	N/A	N/A	1.00	0.00	0.00
18	JHotDraw	N/A	3	0.8	0.40	0.35	0.25	1.00	0.00	0.00
	Tomcat	N/A	3	1.0	0.50	0.30	0.20	0.60	0.20	0.20
	HSQldb	N/A	3	1.2	0.40	0.35	0.25	1.00	0.00	0.00

### Fase de Ordenação de Grupos

Os experimentos para a avaliação da abordagem CAAMPI quanto à ordenação de grupos foram efetuados através da execução do CAAMPI4J com a melhor instância obtida na fase de agrupamento, combinando a execução com cada uma das quatro medidas de ordenação: GSRank, GSIRank, GFRank e GFIRank.

As medidas de ordenação foram avaliadas através dos índices de qualidade de liderança, economia e colocação. A melhor medida de ordenação eleita foi aquela que obteve simultaneamente o maior valor nos três índices de qualidade no sistema avaliado. Caso para um determinado sistema, nenhuma medida tenha obtido simultaneamente o maior valor nos três índices, o critério adotado foi a seleção da medida que obteve a menor diferença de índice para o resultado ótimo (1.00000000), conforme a Equação 6.15.

$$diferenca = (1 - lideranca) + (1 - economia) + (1 - colocacao) \quad (6.15)$$

O valor da diferença de índice para o resultado ótimo é compreendido entre 0 e 3, sendo 0 a obtenção simultânea do resultado ótimo nos índices de qualidade de liderança, economia e colocação. Sendo assim, quanto mais próximo a 0, melhor é o resultado obtido.

### Fase de Identificação de Pontos de Corte

Os experimentos para a avaliação da abordagem CAAMPI quanto à identificação de pontos de corte foram efetuados através da execução do CAAMPI4j com a melhor instância obtida na fase de agrupamento e com o método de identificação de pontos de corte ARPIM.

Os pontos de corte identificados, já com a vinculação dos grupos de candidatos a aspectos, foram avaliados através dos índices de qualidade de cobertura e aproveitamento.

Todos os sistemas foram avaliados quanto ao índice de cobertura, porém, somente o JHotDraw e HSQldb foram avaliados quanto ao índice de aproveitamento. O Tomcat não foi avaliado quanto ao aproveitamento devido a não existência de uma versão orientada a aspectos para confirmação da real aplicação dos pontos de corte.

## 6.2 Resultados e Análise da Fase de Agrupamento

Nesta seção são apresentados os resultados coletados a partir da execução dos experimentos da fase de agrupamento e suas respectivas análises.

Na Tabela 6.4 é apresentado o melhor resultado obtido pelas instâncias para cada um dos sistemas avaliados.

Tabela 6.4: Resultado das instâncias por sistema

Sistema	Instância	Dispersão	Diversificação	Diferença
JHotdraw	1	0.20996693	0.41940639	1.37062668
	2	0.32023809	0.47738095	1.20238096
	3	0.85714285	0.52976190	0.61309525
	4	0.49642857	0.42152777	1.08204366
	5	0.96428571	0.37916666	0.65654763
	6	0.84523809	0.78947368	0.36528823
	7	0.20583213	0.62907407	1.16509380
	8	0.26683673	0.83839918	0.89476409
	9	0.85714285	0.80000000	0.34285715
	10	0.24155844	0.76923076	0.98921080
	11	1.00000000	0.41666666	0.58333334
	12	<u>0.89285714</u>	<u>1.00000000</u>	<u>0.10714286</u>
	13	0.21098550	0.65217391	1.13684059
	14	0.19888451	0.52592592	1.03782911
	15	0.81190476	0.82954545	0.35854979
	16	0.20133477	0.62359550	1.17506973
	17	0.87755102	0.81333333	0.30911565
	18	0.89285714	0.88596491	0.22117795
Tomcat	1	0.74907902	0.43234323	0.81857775
	2	0.75793650	0.43429038	0.80777312
	3	0.97222222	0.58630952	0.44146826
	4	0.78408890	0.51099669	0.70491441
	5	1.00000000	0.43666666	0.56333334
	6	0.98611111	0.62356322	0.39032567
	7	0.66589062	0.69733796	0.63677142
	8	0.71255611	0.84608378	0.44136009
	9	0.91666666	0.90990990	0.17342342
	10	0.72373015	0.80906148	0.46720835
	11	1.00000000	0.42361111	0.57638888
	12	<u>0.95833333</u>	<u>0.91441441</u>	<u>0.12725225</u>
	13	0.71857879	0.75096801	0.53045318
	14	0.69468000	0.51923937	0.78608062
	15	0.85771604	0.87356321	0.26872073
	16	0.74555776	0.63178053	0.62266170
	17	0.88842592	0.75000000	0.36157407
	18	0.90933641	0.86728395	0.22337962
HSQLDB	1	0.86904761	0.50000000	0.63095238
	2	0.84523809	0.48529411	0.66946778
	3	0.91666666	0.52083333	0.56250000
	4	0.92857142	0.44761904	0.62380952
	5	0.96428571	0.46527777	0.57043650
	6	0.96428571	0.56666667	0.46904762
	7	0.83333333	0.92500000	0.24166667
	8	0.83333333	0.89473684	0.27192982
	9	0.86904762	0.94736842	0.18358396
	10	0.85714286	0.83333333	0.30952381
	11	1.00000000	0.47272727	0.52727273
	12	<u>0.96428571</u>	<u>0.86666667</u>	<u>0.16904762</u>
	13	0.84523810	0.72222222	0.43253968
	14	0.83333333	0.77500000	0.39166667
	15	0.86904762	0.72549020	0.40546218
	16	0.96428571	0.47500000	0.56071429
	17	1.00000000	0.56944444	0.43055556
	18	<u>0.96428571</u>	<u>0.86666667</u>	<u>0.16904762</u>

Conforme pode ser observado na Tabela 6.4, a instância 12 obteve o melhor resultado



para os três sistemas avaliados.

Na Tabela 6.5 é apresentada uma visão das 3 instâncias que obtiveram o melhor resultado para cada sistema, enfatizando quais foram as melhores medidas de distância e algoritmos de agrupamento.

Tabela 6.5: Melhores medidas de distância e algoritmos de agrupamento por sistema

Sistema	Posição	Medida	Algoritmo Agrupamento	Instância
JHotDraw	1°	$D_{SOND}$	Hierárquico Clássico	12
	2°	$D_{SOND}$	CHAMELEON	18
	3°	$D_N$	CHAMELEON	17
Tomcat	1°	$D_{SOND}$	Hierárquico Clássico	12
	2°	$D_S$	Hierárquico Clássico	9
	3°	$D_{SOND}$	CHAMELEON	18
HSQLDB	1°	$D_{SOND}$	Hierárquico Clássico	12
	1°	$D_{SOND}$	CHAMELEON	18
	3°	$D_S$	Hierárquico Clássico	9

Conforme observado, as medidas de distância que obtiveram os melhores resultados foram a  $D_{SOND}$ ,  $D_S$  e  $D_N$  respectivamente. Relembrando, a medida  $D_{SOND}$  possui por característica principal a identificação combinada dos sintomas de espalhamento, clonagem e convenções de nomes. Enquanto a medida  $D_S$  tem por característica a identificação do sintoma de espalhamento, incluindo o polimorfismo em sua heurística, e, a medida  $D_N$  tem por característica a identificação do sintoma de convenção de nomes, o que inclui a comparação de semelhança entre nome de métodos, classes, parâmetros e retornos. Quanto aos algoritmos de agrupamento, o Hierárquico Clássico e o CHAMELEON obtiveram respectivamente os melhores resultados.

Na Tabela 6.6 é apresentada uma visão de qual algoritmo de agrupamento obteve o melhor resultado para cada uma das medidas de distância.

Tabela 6.6: Melhor algoritmo de agrupamento por medida de distância

Medida	Sistema	Algoritmo Agrupamento	Instância
$D_S^{CCC}$	JHotDraw	CHAMELEON	13
	Tomcat	CHAMELEON	13
	HSQLDB	Hierárquico Clássico	7
$D^{HBZH}$	JHotDraw	Hierárquico Clássico	8
	Tomcat	Hierárquico Clássico	8
	HSQLDB	Hierárquico Clássico	8
$D_S$	JHotDraw	Hierárquico Clássico	9
	Tomcat	Hierárquico Clássico	9
	HSQLDB	Hierárquico Clássico	9

Continua na próxima página ...

.. continuação da página anterior			
Medida	Sistema	Algoritmo Agrupamento	Instância
$D_O$	JHotDraw	Hierárquico Clássico	10
	Tomcat	Hierárquico Clássico	10
	HSQLDB	Hierárquico Clássico	10
$D_N$	JHotDraw	CHAMELEON	17
	Tomcat	CHAMELEON	17
	HSQLDB	CHAMELEON	17
$D_{SOND}$	JHotDraw	Hierárquico Clássico	12
	Tomcat	Hierárquico Clássico	12
	HSQLDB	Hierárquico Clássico e CHAMELEON	12,18

Conforme observado na Tabela 6.6, os algoritmos de agrupamento baseados em modelo hierárquico foram mais adequados para a fase de agrupamento quando comparados ao algoritmo particional *k-medoids* aplicando as mesmas medidas de distância. Este resultado reforça a conclusão obtida pelos autores Grigoreta et al no estudo [21], em que afirmam que os algoritmos baseados em modelo hierárquico foram os mais eficientes na mineração de aspectos.

Na Tabela 6.7 é apresentada uma visão de qual medida de distância obteve o melhor resultado para cada um dos algoritmos de agrupamento.

Tabela 6.7: Melhor medida de distância por algoritmo de agrupamento

Algoritmo Agrupamento	Sistema	Medida	Instância
Particional	JHotDraw	$D_{SOND}$	6
	Tomcat	$D_{SOND}$	6
	HSQLDB	$D_{SOND}$	6
Hierárquico Clássico	JHotDraw	$D_{SOND}$	12
	Tomcat	$D_{SOND}$	12
	HSQLDB	$D_{SOND}$	12
CHAMELEON	JHotDraw	$D_{SOND}$	18
	Tomcat	$D_{SOND}$	18
	HSQLDB	$D_{SOND}$	18

Conforme observado na Tabela 6.7, a medida de distância  $D_{SOND}$  conseguiu alcançar de forma unânime o melhor resultado em todos os sistemas, independentemente do algoritmo de agrupamento aplicado. O excelente resultado obtido pela medida  $D_{SOND}$  foi devido ao extenso estudo efetuado neste trabalho das formas que os interesses transversais se apresentam na prática nos sistemas orientados a objetos.

Foi observado durante os experimentos que o sintoma que ocorreu em maior quantidade foi o espalhamento. O espalhamento já era explorado pelas medidas  $D_S^{CCC}$  e  $D^{HBZH}$ , porém, foi identificado um fato que faz com que estas medidas não sejam tão eficazes: a vasta utilização de polimorfismo. Conforme observação, muitos interesses transversais

foram implementados através de interfaces e vários métodos que implementam estas interfaces. O uso geralmente ocorreu em dois tipos de categorias:

- Uso da interface para fins de polimorfismo: Vários métodos implementaram uma interface, porém a chamada foi efetuada sempre via interface. Isto fez com que somente o método da interface fosse identificado como interesse transversal, ignorando as implementações.
- Uso da interface para fins de contrato: Vários métodos implementaram a interface para fins de garantir um contrato com determinados métodos, porém os métodos foram chamados de forma pulverizada pelos módulos. Isto fez com que os métodos fossem classificados em grupos diferentes devido ao baixo espalhamento individual e ao conjunto de espalhamento distinto.

A solução proposta foi a captura do polimorfismo na medida  $D_S$ , contabilizando além das chamadas efetuadas ao método, as chamadas aos métodos que este sobrescreveu e as chamadas aos métodos que refinaram este. Esta alteração melhorou consideravelmente a qualidade dos grupos encontrados quando o software em análise aplica padrões de projeto, já que muitos padrões de projetos, como *Adapter*, *Consistent Behavior*, *Command* e etc, utilizam estrutura de herança e interfaces para implementar os interesses transversais, e, portanto, o polimorfismo auxilia na identificação de todos os métodos participantes do interesse transversal, gerando grupos mais coesos e menos dispersos.

Apesar da medida  $D_S$  ter obtido individualmente um bom equilíbrio entre dispersão e a diversidade, esta não conseguiu alcançar índices ótimos. A solução proposta foi a exploração de outros sintomas além do espalhamento.

Um situação identificada é que algumas instâncias de interesses transversais foram implementadas através da cópia completa ou parcial, clonagem, do corpo de um método que é espalhado por vários módulos. Estas cópias do mesmo interesse transversal foram geralmente invocadas por módulos distintos, tornando estes métodos distantes para a medida de espalhamento, o que faz com que estes métodos, mesmo sendo pertencentes ao mesmo interesse transversal, sejam classificados em grupos distintos.

O sintoma de clonagem foi capturado pela medida de distância  $D_O$ , o que torna os métodos que possuem lógica similar próximos para a medida, mesmo no caso em que o espalhamento ocorre em módulos distintos. Porém, esta medida não foi tão eficaz quando aplicada individualmente. Uma deficiência identificada foi que devido a ausência de operações nas interfaces, estas foram classificadas em grupos distintos de suas implementações. Esta medida deve ser aperfeiçoada para eliminar esta ocorrência.

A última situação identificada foi a utilização de convenções de nome por vários padrões de projeto. Este sintoma foi capturado pela medida de distância  $D_N$ , que compara a similaridade de nomes de métodos, nomes de classes, parâmetros e retorno, independentemente do espalhamento ou operações efetuadas por estes. Os resultados indicaram que esta medida obteve os grupos com a menor dispersão entre todas as medidas avaliadas, porém, apresentou alta diversidade. Conforme observações, foi identificada uma deficiência na qual os métodos de um interesse transversal com mesmo nome foram agrupados perfeitamente no mesmo grupo, porém, alguns métodos que coincidentemente possuíam nomes similares, acabaram sendo agrupados no mesmo grupo, mesmo pertencendo a interesses transversais diferentes. Esta medida deve ser aperfeiçoada para reduzir esta ocorrência.

### 6.3 Resultados e Análise da Fase de Ordenação de Grupos

Nesta seção são apresentados os resultados coletados a partir da execução dos experimentos da fase de ordenação de grupos e suas respectivas análises.

As medidas de ordenação foram aplicadas nos grupos identificados pela instância 12 da fase de agrupamento, instância que obteve o melhor resultado nos três sistemas avaliados. Os grupos formados e respectivas pontuações podem ser consultados no Apêndice C.

Na Tabela 6.8 são apresentados os resultados obtidos por cada uma das medidas de ordenação para cada sistema.

Tabela 6.8: Resultados das Medidas de Ordenação de Grupos

Sistema	Medida	Liderança	Economia	Colocação	Diferença
JHotDraw	GSRank	0.41176471	0.60857143	0.90137171	1.07829215
	GSIRank	0.29411765	0.35428571	0.86821510	1.48338154
	GFRank	<u>0.35294118</u>	<u>0.75714286</u>	<u>0.91147017</u>	<u>0.97844579</u>
	GFIRank	0.29411765	0.72000000	0.90372801	1.08215434
Tomcat	GSRank	0.54054054	0.67464115	0.96479800	0.82002031
	GSIRank	<u>0.67567568</u>	<u>0.76714514</u>	<u>0.96786092</u>	<u>0.58931826</u>
	GFRank	0.45945946	0.26634769	0.86821510	1.40597775
	GFIRank	0.54054054	0.55821372	0.95176981	0.94947593
HSQLDB	GSRank	0.20000000	0.01200000	0.74441108	2.04358892
	GSIRank	<u>0.26666667</u>	<u>0.16800000</u>	<u>0.74574575</u>	<u>1.81958758</u>
	GFRank	0.20000000	0.01800000	0.72999666	2.05200334
	GFIRank	0.13333333	0.11600000	0.75522189	1.99544478

Conforme pode ser observado na Tabela 6.8, a medida de ordenação GSIRank obteve o melhor resultado em dois sistemas, enquanto a medida GFRank obteve o melhor resultado em um sistema. De forma a definir a melhor medida, o critério de seleção adotado é o melhor resultado médio, ou seja, a medida que obteve na média a menor Diferença. Na Tabela 6.9 são apresentadas as medidas ordenadas pelo melhor resultado médio, onde: Melhor Resultado é o menor Diferença obtida pela medida, Pior Resultado é a maior Diferença obtida pela medida e Média é a média das Diferenças obtidas nos três sistemas. Na Figura 6.1 o mesmo resultado é representado de forma gráfica, apresentando o intervalo entre o pior e melhor resultado de cada medida, enfatizando a média.

Tabela 6.9: Comparação do resultado das medidas de ordenação

Medida	Melhor Resultado	Pior Resultado	Média	Posição
GSIRank	1,81958758	0,58931826	1,29742913	1°
GSRank	2.04358892	0.82002031	1,31396713	2°
GFIRank	1,99544478	0,94947593	1,34235835	3°
GFRank	2,05200334	0,97844579	1,47880896	4°

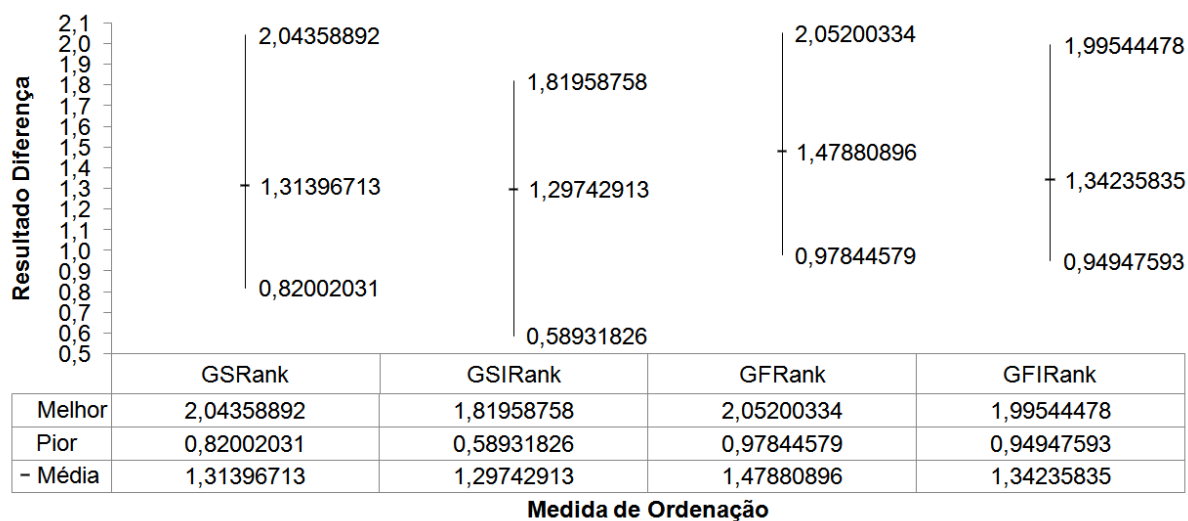


Figura 6.1: Comparação do resultado das medidas de ordenação

Conforme pode ser observado na Tabela 6.9, a medida GSIRank obteve a menor Diferença entre as medidas para o pior caso, a menor Diferença entre as medidas para o melhor caso e a menor média entre as medidas. Portanto, na média, a GSIRank se mostrou ser a medida mais adequada entre as avaliadas para a ordenação de grupos.

Na Tabela 6.10 é apresentada uma visão dos resultados enfatizando a composição do cálculo que é utilizado por cada uma das medidas. A coluna Cálculo da Pontuação informa a heurística utilizada para o cálculo da pontuação, a coluna Interconectividade informa se a medida aplica a interconectividade como redutor, e a coluna Posição informa a colocação obtida pela medida nos resultados (Tabela 6.9).

Tabela 6.10: Visão dos resultados com ênfase nos algoritmos

Medida	Cálculo da Pontuação	Interconectividade	Posição
GSIRank	Soma Espalhamento Distinto do Grupo	Sim	1°
GSRank	Soma Espalhamento Distinto do Grupo	Não	2°
GFIRank	Soma Fan-In Individual Métodos	Sim	3°
GFRank	Soma Fan-In Individual Métodos	Não	4°

Conforme pode ser observado na Tabela 6.10, as medidas que efetuam o cálculo da pontuação baseado na soma do espalhamento distinto do grupo obtiveram respectivamente a 1° e 2° colocação, indicando que este primeiro é mais eficaz do que o cálculo de pontuação baseado na soma do *fan-in* individual dos métodos do grupo.

Também pode ser observado na comparação par a par das medidas baseadas no mesmo tipo de cálculo de pontuação, ou seja, GSIRank contra GSRank e GFIRank contra

GFRank, que o melhor resultado foi obtido pelas medidas que aplicaram a interconectividade como redutor da pontuação. Este resultado indica que a adoção da interconectividade como redutor para penalizar os grupos que possuem métodos pouco relacionados se mostrou eficaz na melhoria dos resultados obtidos.

Um fato identificado na fase de ordenação de grupos, é que os grupos classificados nas primeiras colocações ou são compostos por interesses transversais ou são compostos por métodos utilitários e métodos de acesso a atributos que a fase de filtro não conseguiu eliminar através do processo automatizado. Este fato acaba penalizando as medidas de ordenação haja vista que os métodos utilitários e de acesso possuem alto nível de espalhamento dentro do sistema, e, estes não são considerados interesses transversais. Deve ser estudada uma forma de aperfeiçoar a fase de filtro do CAAMPI4J e/ou da abordagem CAAMPI de forma a criar heurísticas para melhorar a eficácia na eliminação destes métodos de forma automatizada.

## 6.4 Resultados e Análise da Fase de Identificação de Pontos de Corte

Nesta seção são apresentados os resultados coletados a partir da execução dos experimentos da fase de identificação de pontos de corte e suas respectivas análises.

Os pontos de corte identificados foram vinculados aos grupos identificados pela instância 12 da fase de agrupamento, instância que obteve o melhor resultado nos três sistemas avaliados. Os grupos considerados para o cálculo do índice de Cobertura e os aspectos onde foram encontradas as evidências de refatoração para o índice de Aproveitamento podem ser consultados no Apêndice D.

Na Tabela 6.11 é apresentado o resultado obtido pelo método de identificação de pontos de corte ARPIM para os três sistemas avaliados.

Tabela 6.11: Resultado da Identificação de Pontos de Corte com ARPIM

Sistema	Cobertura	Aproveitamento
JHotDraw	0.76470588	0.52631579
Tomcat	0.40540541	N/A
HSQldb	0.40000000	0.66666666

Conforme pode ser observado na Tabela 6.11, a cobertura mínima obtida pelo ARPIM foi de 40%, sendo que o aproveitamento de pontos de corte foi superior a 50% nos 2 casos avaliados. Para o caso específico do JHotDraw, o ARPIM conseguiu identificar pontos de corte na ordem de 3 para cada 4 candidatos a aspectos, sendo que destes, mais da metade foram efetivamente aproveitados no AJHotDraw.

Este resultado demonstra que na prática o ARPIM é atrativo para ser aplicado na mineração de aspectos junto com o agrupamento, pois, o método evidencia as regras existentes envolvendo os métodos de cada candidato a aspecto, facilitando a identificação de como os interesses transversais agem na lógica do sistema.

Um fenômeno identificado nos experimentos é que nem sempre a regra encontrada foi aproveitada como ponto de corte. Em algumas situações a regra era válida, porém, ao

invés de ser refatorada como ponto de corte, foi refatorada pela geração de um adendo ou declaração intertipo contendo a lógica identificada pela regra. Entende-se que desta maneira a solução não foi a ideal, já que o espalhamento não foi eliminado, mas sim, reduzido.

Uma observação importante é que a simples conciliação da versão orientada a objetos com a versão orientada a aspectos não é o ideal para validação do aproveitamento, já que isto está sujeito a habilidade de quem efetuou a refatoração. Bons pontos de corte podem ter sido identificados, porém, podem não ter sido refatorados por questão de escopo. O ideal é alteração da metodologia para que a validação seja efetuada por um especialista no sistema.

## 6.5 Ameaças à Validade

Nesta seção são analisadas algumas possíveis ameaças à validade dos resultados divulgados neste trabalho.

Todas as medidas de avaliação de qualidade propostas foram fundamentadas em métodos totalmente quantitativos, porém, algumas das entradas utilizadas neste trabalho possuem certo nível de inexatidão, ou seja, podem não representar de forma exata a realidade.

As instâncias de interesses transversais consideradas como corretas, validadas, foram baseadas em estudos existentes na literatura e na comparação das versões orientadas a objetos com versões refatoradas para aspectos. Uma ameaça existente é que tanto os estudos quanto as versões refatoradas para aspectos estão sujeitas a subjetividade, seja por um estudo que não considerou todos os aspectos técnicos, seja por uma refatoração para aspectos de forma parcial ou inadequada. Portanto, existe a possibilidade de que boas instâncias de interesses transversais possam não ter sido contabilizadas. O impacto deste fato é que a simples alteração do conjunto de instâncias de interesses transversais pode afetar o resultado obtido nas três fases.

Exemplificando, na fase de agrupamento, alguma medida pode ter sido penalizada por não ter tido a contabilização de algum candidato a aspecto agrupado de forma exata, bem como alguma medida pode ter sido beneficiada por não ter tido a contabilização de algum candidato a aspecto que não foi agrupado de forma correta. Na fase de ordenação, alguma medida pode ter sido penalizada por ter priorizado um interesse transversal que não foi contabilizado, bem como alguma medida pode ter sido beneficiada por não ter priorizado um interesse transversal que não foi contabilizado. Na fase de identificação de pontos de corte, o resultado pode ter sido beneficiado por não ter identificado um ponto de corte corretamente para um interesse transversal que não foi contabilizado, bem como o resultado pode ter sido penalizado por ter identificado corretamente um ponto de corte cujo interesse transversal não foi contabilizado.

Uma ameaça que atinge diretamente o resultado obtido pelas medidas de distância  $D_{SOND}$  e  $D_N$  é o risco de que os fatores de importância adotados não foram adequados, já que não foram testadas todas as possibilidades possíveis de combinação.

Outra ameaça é a dependência existente das fases de ordenação de grupos e identificação de pontos de corte quanto a fase de agrupamento. O efeito já conhecido de dispersão e diversificação que ocorre na fase de agrupamento pode interferir no bom desempenho das medidas de ordenação de grupos e no método de identificação de pontos de corte, beneficiando ou penalizando medidas e resultados.

A fase de ordenação de grupos também foi afetada por uma deficiência da fase de filtro do CAAMPI4J, já que esta não elimina totalmente os métodos de acesso a atributos e utilitários. Devido ao fato de que a abordagem CAAMPI possui a premissa de automação, estes métodos não foram eliminados manualmente. Este fato penaliza o resultado de todas as medidas, e, pode ter interferido na avaliação de quais são as melhores medidas.

Uma questão que atinge pontualmente o método de identificação de pontos de corte é a qualidade das versões orientadas a aspectos dos sistemas orientados a objetos que foram utilizadas como base para afirmar se o ponto de corte foi ou não aproveitado. Um bom ponto de corte identificado pelo método pode não ter sido refatorado na versão orientada a aspectos, penalizando o resultado.

Os sistemas adotados como estudos de caso também introduzem ameaças ao resultado. A qualidade de modelagem da solução e os padrões de codificação adotados por cada um dos sistemas interferem diretamente nos resultados obtidos. Portanto, os resultados obtidos podem se alterar caso outros sistemas sejam avaliados.

## 6.6 Considerações Finais

Neste capítulo foram apresentados os resultados da avaliação da abordagem CAAMPI. Os experimentos englobaram a avaliação dos resultados individuais das fases de agrupamento, ordenação de grupos e identificação de pontos de corte.

Os resultados demonstraram que as medidas propostas neste trabalho obtiveram melhores resultados que as medidas existentes na literatura, considerando os três sistemas avaliados, sendo o melhor resultado obtido pela medida  $D_{SOND}$ .

As medidas de ordenação propostas se mostraram eficientes na ordenação de grupos, já que conseguiram na maioria dos casos reduzir em mais de 75% a quantidade de grupos necessários para análise. A fase de ordenação de grupos se demonstrou estritamente necessária para reduzir o esforço de análise do desenvolvedor.

Já os resultados da fase de identificação de pontos de corte demonstraram que os pontos de corte identificados tiveram mais de 50% de aproveitamento nas versões orientadas a aspectos.

No próximo capítulo é apresentada a conclusão final deste trabalho e as oportunidades identificadas para guiar possíveis trabalhos futuros.



## CAPÍTULO 7

### CONCLUSÃO

Este trabalho apresentou uma abordagem, chamada CAAMPI, para a mineração de aspectos composta por um processo integrado que tem como entrada o código fonte do sistema orientado a objetos e, como saída, os grupos de candidatos a aspectos identificados e respectivos candidatos a pontos de corte.

O processo integrado da abordagem CAAMPI traz várias contribuições para a área de mineração de aspectos. Com o CAAMPI o analista recebe três informações de suma importância: os grupos de candidatos a aspectos, o que indicam quais são os métodos pertencentes a cada interesse transversal, a pontuação obtida por cada grupo, que permite a definição de critérios quantitativos quanto a prioridade de análise dos candidatos a aspectos, e, os pontos de corte identificados, o que provê informações de como estes candidatos a aspectos agem na lógica do sistema.

Para permitir a utilização prática da abordagem proposta, foi implementado o *framework* CAAMPI4J. O CAAMPI4J é uma implementação completa da CAAMPI para a linguagem Java, onde foram implementados os algoritmos de agrupamento *k-medoids*, hierárquico clássico e CHAMELEON. O *framework* serve de modelo de referência para a geração de implementações para outras linguagens de programação.

Quanto às proposições, neste trabalho foram introduzidas quatro novas medidas de distância ( $D_S$ ,  $D_O$ ,  $D_N$  e  $D_{SOND}$ ), três novas medidas de ordenação (GSRank, GSIRank e GFIRank) e um método de identificação de pontos de corte (ARPIM).

A medida de distância  $D_S$  é uma evolução de medidas existentes na literatura, obtendo resultados melhores que as medidas originais. Já as medidas  $D_O$  e  $D_N$  adaptam características de identificação de clones e convenções de nomes, antes somente exploradas como técnicas individuais, em uma técnica baseada em agrupamento. Apesar das medidas  $D_O$  e  $D_N$  não terem obtido resultados tão bons quanto a medida  $D_S$ , elas fazem parte da medida  $D_{SOND}$ , que combina as características de identificação de candidatos a aspectos das medidas  $D_S$ ,  $D_O$  e  $D_N$ . A medida  $D_{SOND}$  se sobressaiu a todas as medidas comparadas, consolidando-se como a melhor medida de distância para a fase de agrupamento em todos os casos avaliados.

Por sua vez, as medidas de ordenação propostas enraizaram o conceito de pontuação de grupos na mineração de aspectos. Esta pontuação permite a ordenação dos grupos de candidatos a aspectos na ordem de impacto que cada grupo gera no sistema em análise. A ordenação viabiliza a definição de critérios quantitativos para a priorização da análise e refatoração dos candidatos a aspectos. As medidas propostas obtiveram bons resultados, estes, que podem ser ainda melhorados através da identificação automática dos métodos utilitários e métodos de acesso, mantendo desta forma somente os grupos de interesses transversais. A introdução da fase de ordenação foi primordial para reduzir o esforço de análise do analista quando na mineração de aspectos utilizando a abordagem CAAMPI.

Por último foi introduzido o método de identificação de pontos de corte ARPIM. O ARPIM permite que a abordagem CAAMPI não esteja limitada a identificação de candidatos a aspectos, mas, que também permita a identificação dos prováveis pontos de corte para os candidatos a aspectos identificados. Os resultados demonstram que existe na prática o aproveitamento dos pontos de corte identificados nas versões refatoradas

para aspectos dos sistemas orientados a objetos. E, mesmo para os pontos de corte que não foram aproveitados, a informação também é válida, já que estes retratam a lógica do sistema em análise, que o analista passa a ter conhecimento sem a necessidade de analisar profundamente o código fonte do sistema.

Sendo assim, os resultados obtidos pelos experimentos confirmam que a abordagem CAAMPI cumpriu com o esperado em sua proposta, trazendo reais avanços para a área de mineração de aspectos.

## 7.1 Trabalhos Futuros

Este trabalho contribui de várias maneiras para a área de mineração de aspectos, porém, este não finda somente com o proposto nesta dissertação. Existem algumas deficiências que foram identificadas nos experimentos que podem ser eliminadas ou amenizadas, bem como, novas abordagens podem ser aplicadas para o aprimoramento da mineração de aspectos. Esta seção é destinada a apresentação de algumas sugestões para o aperfeiçoamento dos estudos propostos neste trabalho.

Uma deficiência relatada na Seção 6.2 referente à medida de distância  $D_O$ , é que esta não consegue comparar interfaces, já que esta medida é baseada em operações, e, as interfaces não possuem operações. É necessário realizar experimentos tratando as interfaces como idênticas a suas implementações e introduzir o polimorfismo, ou seja, inserir uma informação prévia ao algoritmo de agrupamento de que as interfaces e métodos polimórficos devem pertencer ao mesmo grupo. Espera-se que com esta sugestão seja gerado somente um grupo contendo as interfaces e suas implementações, melhorando o resultado da medida.

Outra deficiência relatada na Seção 6.2 é que a medida de distância  $D_N$  acaba agrupando métodos não relacionados no mesmo grupo pelo simples fato de possuírem o mesmo nome. Uma sugestão de melhoria é a definição da dissimilaridade máxima quando for comparado um método polimórfico com um método que não faz parte da mesma hierarquia, mesmo se estes possuírem nomes idênticos. Espera-se com esta sugestão que os métodos pertencentes a uma hierarquia de polimorfismo não sejam agrupados com outros métodos que não fazem parte da mesma hierarquia; e que duas ou mais hierarquias de métodos polimórficos com nomes idênticos sejam agrupadas em grupos respectivos a suas hierarquias.

Em relação à medida  $D_{SOND}$ , esta pode ser aperfeiçoada através do aprimoramento individual das medidas  $D_S$ ,  $D_O$  e  $D_N$  e através de novas abordagens de combinação. Uma sugestão a ser experimentada para uma nova abordagem de combinação é a substituição do cálculo baseado em fatores por um cálculo baseado em gatilhos. Ou seja, caso alguma das medidas obtenha uma similaridade superior a certo limite, o gatilho será disparado, indicando que os métodos devem pertencer ao mesmo grupo. Espera-se com esta sugestão a eliminação dos fatores, e, que cada um dos sintomas possa ser explorado com o mesmo peso, independentemente de um sintoma ocorrer em menor intensidade que os demais.

Quanto a fase de agrupamento, esta também pode ser aperfeiçoada através da exploração de algoritmos de otimização multiobjetivos, que inclusive eliminam a necessidade dos atuais fatores aplicados na medida  $D_{SOND}$ . Além disto, também podem ser aplicados outros tipos de algoritmos de otimização, tais como, Algoritmos Genéticos, Otimização por Nuvem de Partículas, entre outros.

Uma oportunidade identificada é o aprimoramento da abordagem CAAMPI através da introdução de uma fase de catalogação de grupos após a fase de ordenação. Exem-

plificando, caso um grupo possua características conhecidas de um determinado tipo de interesse transversal, este grupo deve ser catalogado como sendo deste tipo específico de interesse transversal. O intuito desta fase é que os grupos sejam catalogados de forma automática em seus respectivos tipos de interesses transversais, ou, que sejam classificados como compostos por métodos utilitários ou de acesso. Espera-se que esta fase traga ainda mais informações ao analista, indicando o tipo de interesse transversal que o grupo representa, e, que seja solucionada a deficiência identificada na avaliação da fase de ordenação de grupos, apresentada na Seção 6.3, onde a fase foi prejudicada devido a presença notável de métodos utilitários e métodos de acesso a atributos. Para tanto, idealizam-se duas possíveis abordagens para serem aplicadas:

- **Processamento de linguagem natural:** Baseado no pressuposto de que os interesses transversais são frequentemente implementados com uso de convenções de nomes, é sugerida a aplicação de uma análise de similaridade semântica dos nomes dos métodos e classes com as implementações comuns de interesses transversais. Por exemplo, se um grupo consiste de métodos com nomes “save”, “persist” ou “insert”, este possui grande tendência a participar do interesse transversal de persistência. Além disso, alguns padrões de projeto usam convenções de sufixos no nome das classes e métodos, como por exemplo os sufixos “DAO” para persistência, “Listener” para observadores e “Command” para operações, o que auxilia a aplicação da abordagem. Métodos utilitários também podem ser identificados através da busca por pacotes, classes ou métodos com a palavra “util” em sua composição. Além disto, o processamento de linguagem natural foi aplicado individualmente no estudo [63], obtendo sucesso, o que indica que a abordagem possui grandes chances de atingir o objetivo esperado.
- **Catálogo de padrões:** É sugerida a execução uma análise de contexto dos métodos pertencentes a cada grupo em busca de padrões característicos de implementação de interesses transversais. Alguns padrões de projeto aplicados em interesses transversais possuem características próprias de implementação, como por exemplo o *Singleton*, que se caracteriza por uma classe com um atributo estático e privado do tipo da própria classe e um método público para capturar a única instância ativa da classe. Esta estrutura é naturalmente criada para encapsular funcionalidades como log e monitoração, que são invocadas por vários módulos do sistema, gerando espalhamento e entrelaçamento. Outra aplicação é a identificação de métodos de acesso a atributos, que possuem por característica a operação de leitura ou escrita a um único atributo de classe. A abordagem é viável, já que catálogos de padrões podem ser encontrados em estudos de refatoração, como em [18] e [46], e aplicável, já que o estudo [19] teve por produto uma ferramenta de mineração de aspectos que identifica candidatos a partir do código fonte com base em um catálogo de padrões. Esta abordagem pode ser aplicada em conjunto com o processamento de linguagem natural de forma a obter melhores resultados.

De forma a efetuar uma mensuração da eficiência da abordagem proposta em comparação com outras técnicas existentes, é interessante que seja efetuada uma avaliação com índices quantitativos que sejam independentes da técnica de mineração de aspectos, ou seja, que permita efetuar comparações com técnicas além da análise de agrupamento. Uma metodologia que permite este tipo de comparação é proposta por Parreira Júnior et al no estudo [51]. No estudo, a metodologia proposta permite a avaliação de diferentes técnicas através da mensuração da precisão na identificação de candidatos a aspectos

válidos e o tempo despendido com a localização dos mesmos por um desenvolvedor. Os índices de qualidade propostos também consideram penalidades para falsos positivos e falsos negativos, o que proporciona a definição de critérios com maior balanceamento, considerando tanto as capacidades de identificação, precisão e tempo gasto com cada técnica.

## BIBLIOGRAFIA

- [1] Agrawal, R., e Srikant, R. Fast Algorithms for Mining Association Rules in Large Databases. *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, páginas 487–499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
- [2] AJHotDraw. Disponível em: <http://ajhotdraw.sourceforge.net/>. Acesso em: 28 abr. 2012.
- [3] AJHSQLDB. Disponível em: <http://ajhsqldb.sourceforge.net/>. Acesso em: 28 abr. 2012.
- [4] Alur, D., Crupi, J., e Malks, D. *Core J2EE Patterns: Best Practices and Design Strategies*. Prentice Hall PTR, 2003.
- [5] AOPSYS. *The JAC Project*. Disponível em: <http://jac.ow2.org>. Acesso em: 28 abr. 2012.
- [6] Apache Software Foundation. *Apache Tomcat*. Disponível em: <http://tomcat.apache.org/>. Acesso em: 28 abr. 2012.
- [7] Barbara, D., e Chen, P. Using the fractal dimension to cluster datasets. *KDD '00: Proceedings of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, páginas 260–264, New York, NY, USA, 2000. ACM Press.
- [8] Baxter, I. D., Yahin, A., Moura, L., Sant'Anna, M., e Bier, L. Clone Detection Using Abstract Syntax Trees. *Proceedings of the International Conference on Software Maintenance, ICSM '98*, Washington, DC, USA, 1998. IEEE Computer Society.
- [9] Breu, S. Aspect mining using event traces. Dissertação de Mestrado, University of Passau, Germany, 2004.
- [10] Breu, S., e Krinke, J. Aspect mining using event traces. *ASE '04: Proceedings of the 19th IEEE International Conference on Automated Software Engineering*, páginas 310–315, Washington, DC, USA, 2004. IEEE Computer Society.
- [11] Centrum, T. T., e Tourwé, T. Mining Aspectual Views using Formal Concept Analysis. In *Proc. of the Fourth IEEE International Workshop on Source Code Analysis and Manipulation - SCAM 2004*, páginas 97–106, Washington, DC, USA, 2004. IEEE Computer Society.
- [12] Czibula, G., Cojocar, G. S., e Czibula, I. G. A partitional clustering algorithm for crosscutting concerns identification. *Proceedings of the 8th WSEAS International Conference on Software Engineering, Parallel and Distributed Systems, SEPADS'09*, páginas 111–116, Stevens Point, Wisconsin, USA, 2009. World Scientific and Engineering Academy and Society (WSEAS).

- [13] Czibula, I. G., Czibula, G., e Cojocar., G. S. Hierarchical Clustering for Identifying Crosscutting Concerns in Object Oriented Software Systems. *INFOCOMP Journal of Computer Science, Universidade Federal de Lavras*, 8(3):21–28, 2009.
- [14] Das, S., Abraham, A., e Konar, A. *Metaheuristic Clustering*. Springer Publishing Company, Incorporated, 1st edition, 2009.
- [15] Dessì, M. *Spring 2.5 Aspect-Oriented Programming*. From technologies to solutions. Packt Publishing Limited, 2009.
- [16] Duan, C., e Cleland-Huang, J. A clustering technique for early detection of dominant and recessive cross-cutting concerns. *Proceedings of the 29th International Conference on Software Engineering Workshops, ICSEW '07*, páginas 89–96, Washington, DC, USA, 2007. IEEE Computer Society.
- [17] Ester, M., Kriegel, H. P., Sander, J., e Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Second International Conference on Knowledge Discovery and Data Mining*, páginas 226–231, Portland, Oregon, 1996. AAAI Press.
- [18] Figueiredo, E., Silva, B., Sant’Anna, C., Garcia, A., Whittle, J., e Nunes, D. Cross-cutting patterns and design stability: An exploratory analysis. *Program Comprehension, 2009. ICPC '09. IEEE 17th International Conference on*, páginas 138–147, Vancouver, BC, 2009. IEEE Computer Society.
- [19] Figueiredo, E., Whittle, J., e Garcia, A. ConcernMorph: metrics-based detection of crosscutting patterns. *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering, ESEC/FSE '09*, páginas 299–300, New York, NY, USA, 2009. ACM.
- [20] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1st edition, 1989.
- [21] Grigoreta, S., Moldovan, C., e Gergely, I. A Comparative Analysis of Clustering Algorithms in Aspect Mining. *Studia Universitatis Babeş-Bolyai, Series Informatica*, 54:75–84, 2009.
- [22] Griswold, B., Kato, Y., e Yuan, J. Aspect Browser: Tool support for managing dispersed aspects. *Proceedings of the 22nd International Conference on Software Engineering. Workshop on Multi-Dimensional Separation of Concerns in Software Engineering, ICSE'2000*, Limerick, Ireland, 2000. ACM.
- [23] Guha, S., Rastogi, R., e Shim, K. Cure: an efficient clustering algorithm for large databases. *Information Systems*, 26(1):35–58, 2001.
- [24] Gybels, K., e Kellens, A. An Experiment in Using Inductive Logic Programming to Uncover Pointcuts. *European Interactive Workshop on Aspects in Software (EIWAS)*, páginas 1–9, Berlin, Germany, 2004.
- [25] Han, J., e Kamber, M. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2nd edition, 2005.

- [26] Hannemann, J., e Kiczales, G. Overcoming the prevalent decomposition in legacy code. *Proceedings of the 23rd International Conference on Software Engineering. Workshop on Advanced Separation of Concerns in Software Engineering, ICSE'2001*, Toronto, Ontario, Canada, 2001. IEEE Press.
- [27] He, L., Bai, H., Zhang, J., e Hu, C. AMUCA Algorithm for Aspect Mining. *Proceedings of Software Engineering and Knowledge Engineering (SEKE)*, páginas 520–524, 2005.
- [28] hMETIS Hypergraph & Circuit Partitioning. Disponível em: <http://glaros.dtc.umn.edu/gkhome/metis/hmetis/overview>. Acesso em: 28 abr. 2012.
- [29] HSQL Development Group. *HSQLDB (HyperSQL DataBase)*. Disponível em: <http://hsqldb.org/>. Acesso em: 28 abr. 2012.
- [30] Huang, J., Lu, Y., e Yang, J. Aspect Mining Using Link Analysis. *Proceedings of the 2010 Fifth International Conference on Frontier of Computer Science and Technology*, páginas 312–317, 2010.
- [31] JHotDraw. Disponível em: <http://www.jhotdraw.org/>. Acesso em: 28 abr. 2012.
- [32] Kamiya, T., Kusumoto, S., e Inoue, K. CCFinder: a multilinguistic token-based code clone detection system for large scale source code. *IEEE Transactions on Software Engineering*, volume 28, páginas 654–670, Piscataway, NJ, USA, 2002. IEEE Press.
- [33] Karypis, G., Han, E.-H. S., e Kumar, V. CHAMELEON: Hierarchical Clustering Using Dynamic Modeling. *IEEE Computer*, volume 32, páginas 68–75, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.
- [34] Kellens, A., Mens, K., e Tonella, P. A survey of automated code-level aspect mining techniques. *Transactions on aspect-oriented software development IV*, páginas 143–162. Springer-Verlag, Berlin, Heidelberg, 2007.
- [35] Kennedy, J., e Eberhart, R. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, volume 4, páginas 1942–1948. IEEE, 1995.
- [36] Kiczales, G., Hilsdale, E., Hugunin, J., Kersten, M., Palm, J., e Griswold, W. G. An Overview of AspectJ. *Proceedings of the 15th European Conference on Object-Oriented Programming, ECOOP '01*, páginas 327–353, London, UK, 2001. Springer-Verlag.
- [37] Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C. V., Loingtier, J.-M., e Irwin, J. Aspect-Oriented Programming. *Proceedings of the 11th European Conference on Object-Oriented Programming, ECOOP '97*, páginas 483–496, London, UK, 1998. Springer-Verlag.
- [38] Kotsiantis, S., e Kanellopoulos, D. Association Rules Mining: A Recent Overview. *GESTS International Transactions on Computer Science and Engineering*, volume 32 of *Global Engineering, Science, and Technology Society*, páginas 71–82, 2006.

- [39] Laddad, R. *AspectJ in Action: Practical Aspect-Oriented Programming*. Manning Publications Co., Greenwich, CT, USA, 2003.
- [40] Levenshtein, V. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707–710, 1966.
- [41] Marin, M. *JHotDraw and Tomcat Fan-In Analysis Results*. Disponível em: <http://swerl.tudelft.nl/bin/view/AMR/FanInAnalysisResults>. Acesso em: 28 abr. 2012.
- [42] Marin, M., Moonen, L., e Deursen, A. v. An Integrated Crosscutting Concern Migration Strategy and its Application to JHotDraw. *Proceedings of the Seventh IEEE International Working Conference on Source Code Analysis and Manipulation*, páginas 101–110, Washington, DC, USA, 2007. IEEE Computer Society.
- [43] Marin, M., van Deursen, A., e Moonen, L. Identifying Aspects Using Fan-In Analysis. *Proceedings of the 11th Working Conference on Reverse Engineering, WCRE '04*, páginas 132–141, Washington, DC, USA, 2004. IEEE Computer Society.
- [44] McLachlan, G. J., e Krishnan, T. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics. Wiley-Interscience, 2nd edition, 2008.
- [45] Mens, K., Poll, B., e González, S. Using intentional source-code views to aid software maintenance. *Proceedings of the International Conference on Software Maintenance, ICSM '03*, páginas 169–178, Washington, DC, USA, 2003. IEEE Computer Society.
- [46] Monteiro, M. P., e Aguiar, A. Patterns for refactoring to aspects: an incipient pattern language. *Proceedings of the 14th Conference on Pattern Languages of Programs, PLOP '07*, páginas 1–7, New York, NY, USA, 2007. ACM.
- [47] Mulder, F., e Zaidman, A. Identifying cross-cutting concerns using software repository mining. *Proceedings of the Joint ERCIM Workshop on Software Evolution (EVOL) and International Workshop on Principles of Software Evolution (IWPSE), IWPSE-EVOL '10*, páginas 23–32, New York, NY, USA, 2010. ACM.
- [48] ORACLE. *Java Compiler Tree API*. Disponível em: <http://download.oracle.com/javase/6/docs/jdk/api/javac/tree/index.html>. Acesso em: 28 abr. 2012.
- [49] ORACLE. *Java Reflection API*. Disponível em: <http://docs.oracle.com/javase/tutorial/reflect/index.html>. Acesso em: 28 abr. 2012.
- [50] Parreira Júnior, P. A., Costa, H. A. X., Camargo, V. V., e Penteado, R. A. D. ComS-CId & DMAsp: Identificação de interesses transversais e recuperação de modelos de classes anotados a partir aplicações OO Java. *Sessão de Ferramentas do Congresso Brasileiro de Software, CBSOFT'2010*, Salvador, BA, Brasil, 2010.
- [51] Parreira Júnior, P. A., Costa, H. A. X., Camargo, V. V., e Penteado, R. A. D. Um estudo da eficiência de técnicas de mineração de interesses. *V Latin American Workshop on Aspect-Oriented Software Development, LA-WASP'2011*, São Paulo, SP, Brasil, 2010.



- [52] Pythius. Disponível em: <http://sourceforge.net/projects/pythius>. Acesso em: 28 abr. 2012.
- [53] Qu, L., e Liu, D. Aspect mining using method call tree. *MUE '07: Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering*, páginas 407–412, Washington, DC, USA, 2007. IEEE Computer Society.
- [54] Robillard, M. P., e Murphy, G. C. Concern graphs: finding and describing concerns using structural program dependencies. *Proceedings of the 24th International Conference on Software Engineering, ICSE '02*, páginas 406–416, New York, NY, USA, 2002. ACM.
- [55] Sampaio, A., Loughran, N., Rashid, A., e Rayson, P. Mining Aspects in Requirements. *Early Aspects 2005: Aspect-Oriented Requirements Engineering and Architecture Design Workshop*, Chicago, Illinois, USA, 2005.
- [56] Serban, G., e Moldovan, G. S. A new genetic clustering based approach in aspect mining. *Proceedings of the 8th World Scientific and Engineering Academy and Society (WSEAS) international conference on Mathematical Methods And Computational Techniques in Electrical Engineering, MMACTEE'06*, páginas 135–140, Stevens Point, Wisconsin, USA, 2006.
- [57] Serban, G., e Moldovan, G. S. A new k-means based clustering algorithm in aspect mining. *Proceedings of the Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC '06*, páginas 69–74, Washington, DC, USA, 2006. IEEE Computer Society.
- [58] Serban, S., e Moldovan, G. A graph algorithm for identification of crosscutting concerns. *Studia Universitatis Babeş-Bolyai, Series Informatica*, volume 51, páginas 53–60, Cluj-Napoca, Romania, 2006. Babeş-Bolyai University.
- [59] Sheikholeslami, G., Chatterjee, S., e Zhang, A. WaveCluster: a multi-resolution clustering approach for very large spatial databases. *Proceedings of the 24rd International Conference on Very Large Data Bases, VLDB '98*, páginas 428–439, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.
- [60] Shepherd, D., Gibson, E., e Pollock, L. L. Design and evaluation of an automated aspect mining tool. *Proceedings of the International Conference on Software Engineering Research and Practice, SERP '04*, páginas 601–607, Las Vegas, Nevada, USA, 2004. CSREA Press.
- [61] Shepherd, D., Palm, J., Pollock, L., e Chu-Carroll, M. Timna: a framework for automatically combining aspect mining analyses. *Proceedings of the 20th IEEE/ACM international conference on Automated Software Engineering, ASE '05*, páginas 184–193, New York, NY, USA, 2005. ACM.
- [62] Shepherd, D., e Pollock, L. Interfaces, aspects, and views. *Linking Aspect Technology and Evolution Workshop, LATE '05*, Chicago, USA, 2005.
- [63] Shepherd, D., Pollock, L., e Tourwé, T. Using language clues to discover crosscutting concerns. *Proceedings of the 2005 workshop on Modeling and Analysis of Concerns in Software, MACS '05*, páginas 1–6, New York, NY, USA, 2005. ACM.

- [64] Spinczyk, O., Lohmann, D., e Urban, M. AspectC++: an AOP extension for C++. *Software Developers Journal*, 5:68–76, 2005.
- [65] Störzer, M., Krinke, J., e Breu, S. Trace analysis for aspect application. *Analysis of Aspect-Oriented Software*, AAOS '03, páginas 1–6, Darmstadt, Germany, 2003.
- [66] Sun Microsystems. *Java BluePrints - J2EE Patterns - MVC*. Disponível em: <http://java.sun.com/blueprints/patterns/MVC-detailed.html>, 2002. Acesso em: 28 abr. 2012.
- [67] Tonella, P., e Ceccato, M. Aspect Mining through the Formal Concept Analysis of Execution Traces. *Proceedings of the 11th Working Conference on Reverse Engineering*, páginas 112–121, Washington, DC, USA, 2004. IEEE Computer Society.
- [68] Wang, W., Yang, J., e Muntz, R. R. STING: A statistical information grid approach to spatial data mining. *Proceedings of the 23rd International Conference on Very Large Data Bases*, VLDB '97, páginas 186–195, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.
- [69] Xu, R., e Wunsch, D. *Clustering*. IEEE Press Series on Computational Intelligence. Wiley-IEEE Press, Piscataway, NJ, USA, 2008.
- [70] Zeng, Y., Hong, Y., Liu, Y., e Wang, J. Maximum tree method based aspect mining method. *Chongqing Daxue Xuebao/Journal of Chongqing University*, 32:1221–1225, 2009.
- [71] Zhang, C., e Jacobsen, H. PRISM is research in aspect mining. *Companion to the 19th annual ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages and Applications*, OOPSLA '04, páginas 20–21, New York, NY, USA, 2004. ACM.
- [72] Zhang, D., Guo, Y., e Chen, X. Automated aspect recommendation through clustering-based fan-in analysis. *Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering*, ASE '08, páginas 278–287, Washington, DC, USA, 2008. IEEE Computer Society.
- [73] Zhang, T., Ramakrishnan, R., e Livny, M. BIRCH: an efficient data clustering method for very large databases. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, SIGMOD '96, páginas 103–114, New York, NY, USA, 1996. ACM.

## APÊNDICE A

### INSTÂNCIAS DE INTERESSES TRANSVERSAIS VALIDADAS

Este apêndice apresenta a lista de instâncias de interesses transversais validadas respectivamente no JHotDraw, Tomcat e HSQLDB. Para os sistemas JHotDraw e HSQLDB também são apresentadas as evidências contendo o mapeamento das instâncias validadas com os respectivos aspectos refactorados.

Na Tabela A.1 são apresentados os métodos que compõem cada uma das instâncias de interesses transversais validadas no sistema JHotDraw.

Tabela A.1: Instâncias de interesses transversais validadas no JHotDraw

Interesse(Instância)	Método
Undo(1)	util.Undoable.undo() util.UndoableAdapter.undo() util.Undoable.isUndoable() util.UndoableAdapter.isUndoable() contrib.PolygonScaleHandle\$UndoActivity.undo() contrib.TextAreaTool\$UndoActivity.undo() contrib.TriangleRotationHandle\$UndoActivity.undo() figures.BorderTool\$UndoActivity.undo() figures.ConnectedTextTool\$UndoActivity.undo() figures.ConnectedTextTool\$DeleteUndoActivity.undo() figures.FontSizeHandle\$UndoActivity.undo() figures.GroupCommand\$UndoActivity.undo() figures.InsertImageCommand\$UndoActivity.undo() figures.PolyLineHandle\$UndoActivity.undo() figures.RadiusHandle\$UndoActivity.undo() figures.TextTool\$UndoActivity.undo() figures.UngroupCommand\$UndoActivity.undo() standard.AlignCommand\$UndoActivity.undo() standard.ResizeHandle\$UndoActivity.undo() standard.ChangeAttributeCommand\$UndoActivity.undo() standard.ChangeConnectionHandle\$UndoActivity.undo() standard.ConnectionTool\$UndoActivity.undo() standard.CutCommand\$UndoActivity.undo() standard.DeleteCommand\$UndoActivity.undo() standard.DragTracker\$UndoActivity.undo() standard.PasteCommand\$UndoActivity.undo() standard.SelectAllCommand\$UndoActivity.undo() standard.SendToBackCommand\$UndoActivity.undo()
Undo(2)	util.Undoable.isRedoable() util.UndoableAdapter.isRedoable() util.Undoable.redo() util.UndoableAdapter.redo() contrib.PolygonScaleHandle\$UndoActivity.redo() contrib.TextAreaTool\$UndoActivity.redo() contrib.TriangleRotationHandle\$UndoActivity.redo() figures.BorderTool\$UndoActivity.redo() figures.ConnectedTextTool\$UndoActivity.redo() figures.FontSizeHandle\$UndoActivity.redo() figures.GroupCommand\$UndoActivity.redo() figures.InsertImageCommand\$UndoActivity.redo() figures.PolyLineHandle\$UndoActivity.redo() figures.RadiusHandle\$UndoActivity.redo() figures.TextTool\$UndoActivity.redo() figures.UngroupCommand\$UndoActivity.redo() standard.AlignCommand\$UndoActivity.redo() standard.ResizeHandle\$UndoActivity.redo() standard.ChangeAttributeCommand\$UndoActivity.redo() standard.ChangeConnectionHandle\$UndoActivity.redo() standard.ConnectionTool\$UndoActivity.redo() standard.CutCommand\$UndoActivity.redo() standard.DeleteCommand\$UndoActivity.redo() standard.DragTracker\$UndoActivity.redo() standard.PasteCommand\$UndoActivity.redo() standard.SelectAllCommand\$UndoActivity.redo() standard.SendToBackCommand\$UndoActivity.redo() figures.ConnectedTextTool\$DeleteUndoActivity.redo()
Composite(1)	framework.DrawingView.add(Figure) standard.StandardDrawing.add(Figure) standard.StandardDrawingView.add(Figure) samples.javadraw.BouncingDrawing.add(Figure) standard.NullDrawingView.add(Figure) framework.Drawing.add(Figure)
Consistent(1)	standard.AbstractCommand.isExecutable() util.Command.isExecutable() util.UndoableCommand.isExecutable()
Consistent(2)	framework.DrawingView.checkDamage() standard.NullDrawingView.checkDamage() standard.StandardDrawingView.checkDamage()

Continua na próxima página ...

... continuação da página anterior	
Interesse(Instância)	Método
Consistent(3)	standard.NullDrawingView.clearSelection() standard.StandardDrawingView.clearSelection() framework.DrawingView.clearSelection()
Command(1)	standard.AbstractCommand.execute() contrib.zoom.ZoomCommand.execute() figures.GroupCommand.execute() figures.InsertImageCommand.execute() figures.UngroupCommand.execute() standard.AlignCommand.execute() standard.BringToFrontCommand.execute() standard.ChangeAttributeCommand.execute() standard.CopyCommand.execute() standard.CutCommand.execute() standard.DeleteCommand.execute() standard.DuplicateCommand.execute() standard.PasteCommand.execute() standard.SelectAllCommand.execute() standard.SendToBackCommand.execute() standard.ToggleGridCommand.execute() util.Command.execute() util.RedoCommand.execute() util.UndoableCommand.execute() util.UndoCommand.execute()
Command(2)	standard.DecoratorFigure.moveBy(int,int) contrib.html.HTMLTextAreaFigure.moveBy(int,int) contrib.TextAreaFigure.moveBy(int,int) figures.TextFigure.moveBy(int,int) framework.Figure.moveBy(int,int) standard.AbstractFigure.moveBy(int,int)
Persistence(1)	util.Storable.write(StorableOutput) standard.AbstractFigure.write(StorableOutput) contrib.TriangleFigure.write(StorableOutput) contrib.TextAreaFigure.write(StorableOutput) figures.TextFigure.write(StorableOutput) contrib.PolygonFigure.write(StorableOutput) figures.ImageFigure.write(StorableOutput) figures.RectangleFigure.write(StorableOutput) figures.EllipseFigure.write(StorableOutput) figures.RoundRectangleFigure.write(StorableOutput) contrib.html.HTMLTextAreaFigure.write(StorableOutput) figures.AttributeFigure.write(StorableOutput)
Persistence(2)	util.Storable.read(StorableInput) standard.AbstractFigure.read(StorableInput) contrib.TriangleFigure.read(StorableInput) figures.ImageFigure.read(StorableInput) contrib.PolygonFigure.read(StorableInput) figures.RectangleFigure.read(StorableInput) figures.EllipseFigure.read(StorableInput) figures.RoundRectangleFigure.read(StorableInput) figures.AttributeFigure.read(StorableInput) contrib.html.HTMLTextAreaFigure.read(StorableInput) contrib.TextAreaFigure.read(StorableInput) figures.TextFigure.read(StorableInput)
Observer(1)	framework.Figure.addFigureChangeListener(FigureChangeListener) framework.Figure.removeFigureChangeListener(FigureChangeListener) standard.AbstractFigure.addFigureChangeListener(FigureChangeListener) standard.AbstractFigure.removeFigureChangeListener(FigureChangeListener)
Observer(2)	framework.Figure.invalidate() figures.NullFigure.invalidate() standard.AbstractFigure.invalidate()
Observer(3)	framework.DrawingView.addFigureSelectionListener(FigureSelectionListener) framework.DrawingView.removeFigureSelectionListener(FigureSelectionListener) standard.NullDrawingView.addFigureSelectionListener(FigureSelectionListener) standard.NullDrawingView.removeFigureSelectionListener(FigureSelectionListener) standard.StandardDrawingView.addFigureSelectionListener(FigureSelectionListener) standard.StandardDrawingView.removeFigureSelectionListener(FigureSelectionListener)
Observer(4)	figures.TextFigure.changed() framework.Figure.changed() framework.Figure.willChange() standard.AbstractFigure.changed() standard.AbstractFigure.willChange()

Na Tabela A.2 é apresentado o mapeamento das instâncias validadas no JHotDraw com os respectivos aspectos refactorados no AJHotDraw.

Tabela A.2: Mapeamento das instâncias validadas do JHotDraw com aspectos refactorados no AJHotDraw

Interesse(Instância)	Aspecto
Undo(1)	*Undo.aj
Undo(2)	*Undo.aj
Composite(1)	GroupCommandUndo.aj
Consistent(1)	UndoableCommand.aj
Consistent(2)	CommandContracts.aj
Consistent(3)	*Undo.aj
Command(1)	CommandContracts.aj, *CommandUndo.aj
Command(2)	AlignCommand.aj
Observer(1)	PersistentTextFigure.aj
Observer(2)	SelectionChangedNotification.aj
Observer(3)	SelectionChangedNotification.aj, FigureSelectionSubjectRole.aj
Observer(4)	PersistentTextFigure.aj
Persistence(1)	Persistence*.aj
Persistence(2)	Persistence*.aj

Na Tabela A.3 são apresentados os métodos que compõem cada uma das instâncias de interesses transversais validadas no sistema Tomcat.

Tabela A.3: Instâncias de interesses transversais validadas no Tomcat

Interesse(Instância)	Método
LifeCycle(1)	catalina.core.StandardEngine.stop() catalina.core.StandardWrapper.stop() catalina.ha.context.ReplicatedContext.stop() catalina.core.ContainerBase.stop() catalina.core.StandardContext.stop() catalina.startup.Catalina.stop() catalina.core.StandardPipeline.stop() catalina.core.StandardService.stop() catalina.core.StandardServer.stop() catalina.startup.Embedded.stop() catalina.authenticator.SSLAuthenticator.stop() catalina.loader.WebappClassLoader.stop() catalina.session.JDBCStore.stop() catalina.cluster.session.SimpleTcpReplicationManager.stop() catalina.ha.session.BackupManager.stop() catalina.ha.session.SimpleTcpReplicationManager.stop() catalina.session.StandardManager.stop() catalina.session.PersistentManagerBase.stop() catalina.cluster.session.DeltaManager.stop() catalina.ha.session.DeltaManager.stop() catalina.connector.Connector.stop() catalina.session.StoreBase.stop() catalina.valves.FastCommonAccessLogValve.stop() catalina.authenticator.AuthenticatorBase.stop() catalina.authenticator.SingleSignOn.stop() catalina.valves.SemaphoreValve.stop() catalina.cluster.session.JvmRouteBinderValve.stop() catalina.ha.session.JvmRouteBinderValve.stop() catalina.valves.ExtendedAccessLogValve.stop() catalina.valves.AccessLogValve.stop() catalina.valves.JDBCAccessLogValve.stop() catalina.loader.WebappLoader.stop() catalina.cluster.tcp.SimpleTcpCluster.stop() catalina.ha.tcp.SimpleTcpCluster.stop() catalina.Lifecycle.stop() catalina.realm.RealmBase.stop() catalina.realm.JNDIRealm.stop() catalina.realm.JDBCRealm.stop() catalina.realm.MemoryRealm.stop() catalina.realm.UserDatabaseRealm.stop() catalina.realm.DataSourceRealm.stop() catalina.realm.JAASRealm.stop()
LifeCycle(2)	catalina.core.StandardEngine.start() catalina.core.StandardWrapper.start() catalina.ha.context.ReplicatedContext.start() catalina.core.ContainerBase.start() catalina.core.StandardContext.start() catalina.startup.Catalina.start() catalina.core.StandardPipeline.start() catalina.core.StandardService.start() catalina.core.StandardServer.start() catalina.startup.Embedded.start() catalina.authenticator.SSLAuthenticator.start() catalina.loader.WebappClassLoader.start() catalina.session.JDBCStore.start() catalina.cluster.session.SimpleTcpReplicationManager.start() catalina.ha.session.BackupManager.start() catalina.ha.session.SimpleTcpReplicationManager.start() catalina.session.StandardManager.start() catalina.session.PersistentManagerBase.start() catalina.cluster.session.DeltaManager.start() catalina.ha.session.DeltaManager.start() catalina.connector.Connector.start() catalina.session.StoreBase.start() catalina.valves.FastCommonAccessLogValve.start() catalina.authenticator.AuthenticatorBase.start() catalina.authenticator.SingleSignOn.start() catalina.valves.SemaphoreValve.start() catalina.cluster.session.JvmRouteBinderValve.start() catalina.ha.session.JvmRouteBinderValve.start() catalina.valves.ExtendedAccessLogValve.start() catalina.valves.AccessLogValve.start() catalina.valves.JDBCAccessLogValve.start() catalina.loader.WebappLoader.start() catalina.cluster.tcp.SimpleTcpCluster.start() catalina.ha.tcp.SimpleTcpCluster.start() catalina.Lifecycle.start() catalina.realm.RealmBase.start() catalina.realm.JNDIRealm.start() catalina.realm.JDBCRealm.start() catalina.realm.MemoryRealm.start() catalina.realm.UserDatabaseRealm.start() catalina.realm.DataSourceRealm.start() catalina.realm.JAASRealm.start()

Continua na próxima página ...

... continuação da página anterior	
Interesse(Instância)	Método
Observer(1)	catalogina.Lifecycle.addListener(LifecycleListener)
	catalogina.loader.WebappClassLoader.addListener(LifecycleListener)
	catalogina.core.ContainerBase.addListener(LifecycleListener)
	catalogina.core.StandardServer.addListener(LifecycleListener)
	catalogina.core.StandardPipeline.addListener(LifecycleListener)
	catalogina.core.StandardService.addListener(LifecycleListener)
	catalogina.authenticator.SingleSignOn.addListener(LifecycleListener)
	catalogina.connector.Connector.addListener(LifecycleListener)
	catalogina.session.StandardManager.addListener(LifecycleListener)
	catalogina.cluster.session.DeltaManager.addListener(LifecycleListener)
	catalogina.cluster.tcp.SimpleTcpCluster.addListener(LifecycleListener)
	catalogina.ha.tcp.SimpleTcpCluster.addListener(LifecycleListener)
	catalogina.loader.WebappLoader.addListener(LifecycleListener)
	catalogina.startup.Embedded.addListener(LifecycleListener)
	catalogina.realm.RealmBase.addListener(LifecycleListener)
	catalogina.session.StoreBase.addListener(LifecycleListener)
	catalogina.authenticator.AuthenticatorBase.addListener(LifecycleListener)
	catalogina.session.PersistentManagerBase.addListener(LifecycleListener)
	catalogina.valves.FastCommonAccessLogValve.addListener(LifecycleListener)
	catalogina.valves.ExtendedAccessLogValve.addListener(LifecycleListener)
	catalogina.valves.AccessLogValve.addListener(LifecycleListener)
	catalogina.valves.JDBCAccessLogValve.addListener(LifecycleListener)
	catalogina.valves.SemaphoreValve.addListener(LifecycleListener)
catalogina.cluster.session.JvmRouteBinderValve.addListener(LifecycleListener)	
catalogina.ha.session.JvmRouteBinderValve.addListener(LifecycleListener)	
catalogina.util.LifecycleSupport.addListener(LifecycleListener)	
Observer(2)	catalogina.core.ContainerBase.fireContainerEvent(String, Object)
Observer(3)	catalogina.util.LifecycleSupport.fireLifecycleEvent(String, Object)
Observer(4)	catalogina.util.LifecycleSupport.removeLifecycleListener(LifecycleListener)
Contract(1)	catalogina.Session.isValid()
	catalogina.session.StandardSession.isValid()
	catalogina.cluster.session.DeltaSession.isValid()
Contract(2)	catalogina.ha.session.DeltaSession.isValid()
Contract(3)	catalogina.security.SecurityUtil.isPackageProtectionEnabled()
Contract(3)	catalogina.connector.Response.isCommitted()
	catalogina.connector.ResponseFacade.isCommitted()
Contract(4)	catalogina.storeconfig.StoreRegistry.findDescription(String)
Chain(1)	catalogina.Valve.invoke(Request, Response)
	catalogina.valves.ValveBase.invoke(Request, Response)
	catalogina.core.StandardWrapperValve.invoke(Request, Response)
	catalogina.core.StandardContextValve.invoke(Request, Response)
	catalogina.core.StandardEngineValve.invoke(Request, Response)
	catalogina.core.StandardHostValve.invoke(Request, Response)
	catalogina.valves.PersistentValve.invoke(Request, Response)
	catalogina.cluster.tcp.ReplicationValve.invoke(Request, Response)
	catalogina.ha.tcp.ReplicationValve.invoke(Request, Response)
	catalogina.cluster.session.JvmRouteBinderValve.invoke(Request, Response)
	catalogina.ha.session.JvmRouteBinderValve.invoke(Request, Response)
	catalogina.valves.ErrorReportValve.invoke(Request, Response)
	catalogina.valves.SemaphoreValve.invoke(Request, Response)
	catalogina.valves.RequestDumperValve.invoke(Request, Response)
	catalogina.valves.ExtendedAccessLogValve.invoke(Request, Response)
	catalogina.valves.JDBCAccessLogValve.invoke(Request, Response)
	catalogina.valves.AccessLogValve.invoke(Request, Response)
	catalogina.valves.FastCommonAccessLogValve.invoke(Request, Response)
	catalogina.authenticator.AuthenticatorBase.invoke(Request, Response)
	catalogina.authenticator.SingleSignOn.invoke(Request, Response)
	catalogina.valves.RequestFilterValve.invoke(Request, Response)
catalogina.valves.RemoteAddrValve.invoke(Request, Response)	
catalogina.valves.RemoteHostValve.invoke(Request, Response)	
Command(1)	catalogina.ant.AbstractCatalinaTask.execute()
	catalogina.ant.AbstractCatalinaTask.execute(java.lang.String)
	catalogina.ant.StartTask.execute()
	catalogina.ant.StopTask.execute()
	catalogina.ant.ReloadTask.execute()
	catalogina.ant.SessionsTask.execute()
	catalogina.ant.ServerinfoTask.execute()
	catalogina.ant.UndeployTask.execute()
	catalogina.ant.InstallTask.execute()
	catalogina.ant.ListTask.execute()
	catalogina.ant.JMXQueryTask.execute()
	catalogina.ant.JMXGetTask.execute()
	catalogina.ant.JMXSetTask.execute()
	catalogina.ant.RolesTask.execute()
	catalogina.ant.RemoveTask.execute()
	catalogina.ant.ResourcesTask.execute()
catalogina.ant.DeployTask.execute()	
catalogina.ant.JKStatusUpdateTask.execute()	
Consistent(1)	catalogina.mbeans.MBeanUtils.createManagedName(Object)
Visitor(1)	jasper.compiler.Node\$Visitor.visitBody(Node)
	jasper.compiler.SmapUtil\$SmapGenVisitor.visitBody(Node)
Visitor(2)	jasper.compiler.TextOptimizer\$TextCatVisitor.visitBody(Node)
Visitor(3)	jasper.compiler.Node\$Nodes.visit(Node\$Visitor)
	jasper.compiler.Validator\$ValidateVisitor\$NamedAttributeVisitor.doVisit(Node)
	jasper.compiler.ErrorDispatcher\$ErrorVisitor.doVisit(Node)
	jasper.compiler.SmapUtil\$PreScanVisitor.doVisit(Node)
Composite(1)	jasper.compiler.Node\$Visitor.doVisit(Node)
	jasper.compiler.TextOptimizer\$TextCatVisitor.doVisit(Node)
Composite(1)	catalogina.Container.findChild(String)
	catalogina.core.ContainerBase.findChild(String)

Continua na próxima página ...

... continuação da página anterior	
Interesse(Instância)	Método
Composite(2)	catalina.core.StandardContext.addChild(Container) catalina.core.StandardWrapper.addChild(Container) catalina.core.StandardEngine.addChild(Container) catalina.core.StandardHost.addChild(Container) catalina.Container.addChild(Container) catalina.core.ContainerBase.addChild(Container)
Composite(3)	catalina.Container.removeChild(Container) catalina.core.ContainerBase.removeChild(Container) catalina.core.StandardContext.removeChild(Container)
Composite(4)	catalina.Container.findChildren() catalina.core.ContainerBase.findChildren()
Consistent(2)	catalina.core.ApplicationContextFacade.doPrivileged(ApplicationContext,Object) catalina.core.ApplicationContextFacade.doPrivileged(String,Object)
Consistent(3)	naming.SelectorContext.parseName(String) naming.SelectorContext.parseName(Name)
Consistent(4)	naming.resources.ProxyDirContext.parseName(Name) naming.resources.ProxyDirContext.parseName(String)
Consistent(5)	naming.resources.ProxyDirContext.cacheUnload(String)
Consistent(6)	jasper.compiler.Dumper\$DumpVisitor.printString(String,char,String)
Consistent(6)	jasper.compiler.Dumper\$DumpVisitor.printString(String)
Consistent(7)	jasper.compiler.Dumper\$DumpVisitor.printAttributes(String,Attributes,String)
Consistent(8)	jasper.compiler.Dumper\$DumpVisitor.dumpBody(Node)
Consistent(9)	catalina.util.RequestUtil.filter(String)
Consistent(10)	jasper.security.SecurityUtil.isPackageProtectionEnabled()
Consistent(11)	jasper.compiler.SmapUtil\$SmapGenVisitor.doSmap(Node)
Consistent(12)	jasper.compiler.Parser.parseOptionalBody(Node,String,String)
Consistent(13)	jasper.compiler.Parser.parseAttributes()
Consistent(14)	jasper.compiler.PageDataImpl\$SecondPassVisitor.appendTag(Node)
Consistent(15)	jasper.compiler.JspUtil.checkAttributes(String,Node,ValidAttribute,ErrorDispatcher)
Consistent(16)	jasper.compiler.ErrorDispatcher.dispatch(Mark,String,Object,Exception)
Redirector(1)	jasper.util.SystemLogHandler.findStream()

Na Tabela A.4 são apresentados os métodos que compõem cada uma das instâncias de interesses transversais validadas no sistema HSQldb.

Tabela A.4: Instâncias de interesses transversais validas no HSQldb

Interesse(Instância)	Método
Precondition(1)	jdbc.jdbcConnection.checkClosed() jdbc.jdbcResultSet.checkClosed() jdbc.jdbcStatement.checkClosed()
Exception(1)	Trace.error(int,Object) Trace.error(int,int,Object)
Exception(2)	jdbc.Util.sqlException(HsqlException) jdbc.Util.sqlException(int) jdbc.Util.sqlException(int,String)
Exception(3)	util.CommonSwing.errorMessage(Exception)
Authorization(1)	Grantee.isAdmin()
Authorization(2)	Grantee.isAccessible(Object,int)
Access Control(1)	Session.isReadOnly() SessionInterface.isReadOnly()
Access Control(2)	Database.isFilesReadOnly()
Access Control(3)	Table.enforceNullConstraints(Object) Table.enforceFieldValueLimits(Object,int)
Access Control(4)	Table.isEmpty(Session)
Access Control(5)	Table.isView()
Access Control(6)	Table.isDataReadOnly()
Concurrent(1)	util.DatabaseManagerSwing.backgroundIt(Runnable,String)
Tracing(1)	Trace.printSystemOut(String)

Na Tabela A.5 é apresentado o mapeamento das instâncias validadas no HSQldb com os respectivos aspectos refactorados no AJHSQldb.

Tabela A.5: Mapeamento das instâncias validadas do HSQldb com aspectos refactorados no AJHSQldb

Interesse(Instância)	Aspecto
Precondition(1)	jdbcConnection.PreconditionAspect
Exception(1)	jdbcResultSet.ExceptionHandlingAspect
Exception(2)	jdbcStatement.ExceptionHandlingAspect
Exception(3)	DatabaseManagerSwing.SwingThreadSafetyAspect
Authorization(1)	DatabaseCommandInterpreter.AuthorizationAspect
Authorization(2)	Parser.AuthorizationAspect
Access Control(1)	DatabaseCommandInterpreter.AccessControlAspect
Access Control(2)	CompiledStatementExecutor.AccessControlAspect
Access Control(3)	Table.AccessControlAspect
Access Control(4)	TableWorks.AccessControlAspect
Access Control(5)	Parser.AccessControlAspect
Access Control(6)	Parser.AccessControlAspect
Concurrent(1)	DatabaseManagerSwing.SwingThreadSafetyAspect
Tracing(1)	TracingRedirectOutputAspect

## APÊNDICE B

### CALIBRAÇÃO DAS INSTÂNCIAS DE AGRUPAMENTO

Neste apêndice são apresentadas as tabelas contendo os resultados obtidos para os valores de parâmetros usados nas instâncias dos experimentos efetuados no Capítulo 6, para os três sistemas utilizados: JHotDraw, Tomcat e HSQLDB.

Tabela B.1: Calibração da Instância 1: JHotDraw

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.24633065	0.34031007	1.41335928
2	0.90	0.23160800	0.37316384	1.39522816
3	0.80	0.21372887	0.40880952	1.37746161
4	0.70	0.20996693	0.41940639	1.37062668

Tabela B.2: Calibração da Instância 1: Tomcat

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.71218072	0.35985915	0.92796013
2	0.90	0.72021581	0.38008130	0.89970289
3	0.80	0.73520887	0.42626262	0.83852851
4	0.70	0.74907902	0.43234323	0.81857775

Tabela B.3: Calibração da Instância 1: HSQLDB

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.84523809	0.44047619	0.71428571
2	0.90	0.84523809	0.49074074	0.66402116
3	0.80	0.86904761	0.50000000	0.63095238
4	0.70	0.84523809	0.50980392	0.64495798

Tabela B.4: Calibração da Instância 2: JHotDraw

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.30901360	0.47983630	1.21115010
2	0.90	0.32023809	0.47738095	1.20238096
3	0.80	0.33469387	0.43806689	1.22723924
4	0.70	0.30425170	0.43458485	1.26116345

Tabela B.5: Calibração da Instância 2: Tomcat

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.59424603	0.46891534	0.93683863
2	0.90	0.75744047	0.43338793	0.80917160
3	0.80	0.74404761	0.41473023	0.84122216
4	0.70	0.75793650	0.43429038	0.80777312

Tabela B.6: Calibração da Instância 2: HSQLDB

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.73809523	0.50000000	0.76190476
2	0.90	0.84523809	0.48529411	0.66946778
3	0.80	0.83333333	0.48611111	0.68055555
4	0.70	0.84523809	0.48039215	0.67436974

Tabela B.7: Calibração da Instância 3: JHotDraw

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.72023809	0.43627450	0.84348741
2	0.90	0.73452380	0.47916666	0.78630954
3	0.80	0.85714285	0.52976190	0.61309525
4	0.70	0.85714285	0.52380952	0.61904763

Tabela B.8: Calibração da Instância 3: Tomcat

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.88888888	0.45061728	0.66049384
2	0.90	0.91666666	0.45535714	0.62797620
3	0.80	0.94444444	0.50000000	0.55555556
4	0.70	0.97222222	0.58630952	0.44146826

Tabela B.9: Calibração da Instância 3: HSQLDB

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.89285714	0.44871794	0.65842490
2	0.90	0.86904761	0.46875000	0.66220238
3	0.80	0.88095238	0.52941176	0.58963585
4	0.70	0.91666666	0.52083333	0.56250000



Tabela B.10: Calibração da Instância 4: JHotDraw

n°	distMin	Dispersão	Diversificação	Diferença
1	1.00	0.49642857	0.42152777	1.08204366
2	0.90	0.37976190	0.41363636	1.20660174
3	0.80	0.36250000	0.46720430	1.17029570
4	0.70	0.34506052	0.50155038	1.15338910

Tabela B.11: Calibração da Instância 4: Tomcat

n°	distMin	Dispersão	Diversificação	Diferença
1	1.00	0.55159932	0.44920634	0.99919434
2	0.90	0.79123093	0.43530303	0.77346604
3	0.80	0.78459595	0.48074342	0.73466063
4	0.70	0.78408890	0.51099669	0.70491441

Tabela B.12: Calibração da Instância 4: HSQLDB

n°	distMin	Dispersão	Diversificação	Diferença
1	1.00	0.50000000	0.46875000	1.03125000
2	0.90	0.92857142	0.44761904	0.62380952
3	0.80	0.89285714	0.47023809	0.63690476
4	0.70	0.89285714	0.47023809	0.63690476

Tabela B.13: Calibração da Instância 5: JHotDraw

n°	distMin	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
1	1.00	1.00	0.00	0.00	0.92857142	0.26041666	0.81101192
2	1.00	0.50	0.40	0.10	0.92857142	0.19999999	0.87142859
3	1.00	0.50	0.30	0.20	0.92857142	0.19999999	0.87142859
4	1.00	0.60	0.20	0.20	0.92857142	0.19999999	0.87142859
5	1.00	0.40	0.40	0.20	0.92857142	0.19999999	0.87142859
6	0.90	1.00	0.00	0.00	0.92857142	0.26041666	0.81101192
7	0.90	0.50	0.40	0.10	0.84523809	0.23849206	0.91626985
8	0.90	0.50	0.30	0.20	0.88095238	0.23958333	0.87946429
9	0.90	0.60	0.20	0.20	0.84523809	0.23363095	0.92113096
10	0.90	0.40	0.40	0.20	0.92857142	0.17037037	0.90105821
11	0.80	1.00	0.00	0.00	0.96428571	0.37916666	0.65654763
12	0.80	0.50	0.40	0.10	0.58333333	0.37568027	1.04098640
13	0.80	0.50	0.30	0.20	0.73214285	0.32878787	0.93906928
14	0.80	0.60	0.20	0.20	0.57142857	0.33968253	1.08888890
15	0.80	0.40	0.40	0.20	0.50595238	0.29393939	1.20010823
16	0.70	1.00	0.00	0.00	0.85714285	0.42361111	0.71924604
17	0.70	0.50	0.40	0.10	0.67857142	0.41145833	0.90997025
18	0.70	0.50	0.30	0.20	0.76785714	0.44907407	0.78306879
19	0.70	0.60	0.20	0.20	0.77380952	0.44607843	0.78011205
20	0.70	0.40	0.40	0.20	0.77976190	0.44607843	0.77415967

Tabela B.14: Calibração da Instância 5: Tomcat

n°	distMin	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
1	1.00	1.00	0.00	0.00	1.00000000	0.21774891	0.78225109
2	1.00	0.50	0.40	0.10	0.97222222	0.07812500	0.94965278
3	1.00	0.50	0.30	0.20	1.00000000	0.11515151	0.88484849
4	1.00	0.60	0.20	0.20	0.98611111	0.09848484	0.91540405
5	1.00	0.40	0.40	0.20	1.00000000	0.11515151	0.88484849
6	0.90	1.00	0.00	0.00	1.00000000	0.25703463	0.74296537
7	0.90	0.50	0.40	0.10	0.91203703	0.17895768	0.90900529
8	0.90	0.50	0.30	0.20	0.90277777	0.18059440	0.91662783
9	0.90	0.60	0.20	0.20	0.94444444	0.24455128	0.81100428
10	0.90	0.40	0.40	0.20	0.88888888	0.18010033	0.93101079
11	0.80	1.00	0.00	0.00	1.00000000	0.34791666	0.65208334
12	0.80	0.50	0.40	0.10	0.83179012	0.29916666	0.86904322
13	0.80	0.50	0.30	0.20	0.85648148	0.32936507	0.81415345
14	0.80	0.60	0.20	0.20	0.88194444	0.30701058	0.81104498
15	0.80	0.40	0.40	0.20	0.76504629	0.32693236	0.90802135
16	0.70	1.00	0.00	0.00	1.00000000	0.43666666	0.56333334
17	0.70	0.50	0.40	0.10	0.84887566	0.43728070	0.71384364
18	0.70	0.50	0.30	0.20	0.82870370	0.44909909	0.72219721
19	0.70	0.60	0.20	0.20	0.84814814	0.38620689	0.76564497
20	0.70	0.40	0.40	0.20	0.83101851	0.44594594	0.72303555

Tabela B.15: Calibração da Instância 5: HSQLDB

n°	distMin	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
1	1.00	1.00	0.00	0.00	0.96428571	0.18055555	0.85515873
2	1.00	0.50	0.40	0.10	1.00000000	0.28571428	0.71428571
3	1.00	0.50	0.30	0.20	0.96428571	0.12142857	0.91428571
4	1.00	0.60	0.20	0.20	0.96428571	0.12142857	0.91428571
5	1.00	0.40	0.40	0.20	0.96428571	0.11805555	0.91765873
6	0.90	1.00	0.00	0.00	1.00000000	0.22500000	0.77500000
7	0.90	0.50	0.40	0.10	0.96428571	0.29523809	0.74047619
8	0.90	0.50	0.30	0.20	0.92857142	0.28333333	0.78809523
9	0.90	0.60	0.20	0.20	0.96428571	0.28125000	0.75446428
10	0.90	0.40	0.40	0.20	0.96428571	0.25000000	0.78571428
11	0.80	1.00	0.00	0.00	0.96428571	0.40740740	0.62830687
12	0.80	0.50	0.40	0.10	0.92857142	0.42000000	0.65142857
13	0.80	0.50	0.30	0.20	0.92857142	0.34523809	0.72619047
14	0.80	0.60	0.20	0.20	0.91666666	0.42333333	0.66000000
15	0.80	0.40	0.40	0.20	0.85714285	0.38333333	0.75952380

Continua na próxima página ...







... continuação da página anterior										
n°	<i>distMin</i>	$F_S$	$F_O$	$F_N$	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
57	0.80	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.50000000	0.50000000
58	0.80	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.52564103	0.47435897
59	0.80	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.52564103	0.47435897
60	0.80	0.50	0.30	0.20	0.40	0.40	0.20	0.96428571	0.53333333	0.50238095
61	0.80	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.52380952	0.51190476
62	0.80	0.45	0.45	0.10	0.50	0.40	0.10	0.89285714	0.52941176	0.57773109
63	0.80	0.45	0.45	0.10	0.50	0.30	0.20	0.96428571	0.52380952	0.51190476
64	0.80	0.45	0.45	0.10	0.60	0.20	0.20	0.92857143	0.52222222	0.54920635
65	0.80	0.45	0.45	0.10	0.40	0.40	0.20	0.89285714	0.55882353	0.54831933
66	0.80	0.35	0.35	0.30	1.00	0.00	0.00	0.92857143	0.48214286	0.58928571
67	0.80	0.35	0.35	0.30	0.50	0.40	0.10	0.96428571	0.47435897	0.56135531
68	0.80	0.35	0.35	0.30	0.50	0.30	0.20	0.91666667	0.48333333	0.60000000
69	0.80	0.35	0.35	0.30	0.60	0.20	0.20	0.96428571	0.52380952	0.51190476
70	0.80	0.35	0.35	0.30	0.40	0.40	0.20	0.91666667	0.48958333	0.59375000
71	0.80	0.40	0.35	0.25	1.00	0.00	0.00	0.96428571	0.47435897	0.56135531
72	0.80	0.40	0.35	0.25	0.50	0.40	0.10	0.91666667	0.51111111	0.57222222
73	0.80	0.40	0.35	0.25	0.50	0.30	0.20	0.96428571	0.52380952	0.51190476
74	0.80	0.40	0.35	0.25	0.60	0.20	0.20	0.92857143	0.52222222	0.54920635
75	0.80	0.40	0.35	0.25	0.40	0.40	0.20	0.91666667	0.51666667	0.56666667
76	0.70	0.40	0.40	0.20	1.00	0.00	0.00	0.92857143	0.51190476	0.55952381
77	0.70	0.40	0.40	0.20	0.50	0.40	0.10	0.96428571	0.45833333	0.57738095
78	0.70	0.40	0.40	0.20	0.50	0.30	0.20	0.92857143	0.52222222	0.54920635
79	0.70	0.40	0.40	0.20	0.60	0.20	0.20	0.96428571	0.48809524	0.54761905
80	0.70	0.40	0.40	0.20	0.40	0.40	0.20	0.92857143	0.48888889	0.58253968
81	0.70	0.50	0.30	0.20	1.00	0.00	0.00	0.92857143	0.47619048	0.59523810
82	0.70	0.50	0.30	0.20	0.50	0.40	0.10	0.92857143	0.52222222	0.54920635
83	0.70	0.50	0.30	0.20	0.50	0.30	0.20	0.96428571	0.50000000	0.53571429
84	0.70	0.50	0.30	0.20	0.60	0.20	0.20	0.96428571	0.50000000	0.53571429
85	0.70	0.50	0.30	0.20	0.40	0.40	0.20	0.92857143	0.53125000	0.54017857
86	0.70	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.48809524	0.54761905
87	0.70	0.45	0.45	0.10	0.50	0.40	0.10	0.96428571	0.47435897	0.56135531
88	0.70	0.45	0.45	0.10	0.50	0.30	0.20	0.96428571	0.48809524	0.54761905
89	0.70	0.45	0.45	0.10	0.60	0.20	0.20	0.96428571	0.48809524	0.54761905
90	0.70	0.45	0.45	0.10	0.40	0.40	0.20	0.92857143	0.55555556	0.51587302
91	0.70	0.35	0.35	0.30	1.00	0.00	0.00	1.00000000	0.52564103	0.47435897
92	0.70	0.35	0.35	0.30	0.50	0.40	0.10	0.96428571	0.51282051	0.52289377
93	0.70	0.35	0.35	0.30	0.50	0.30	0.20	0.96428571	0.52380952	0.51190476
94	0.70	0.35	0.35	0.30	0.60	0.20	0.20	0.96428571	0.48809524	0.54761905
95	0.70	0.35	0.35	0.30	0.40	0.40	0.20	0.92857143	0.51190476	0.55952381
96	0.70	0.40	0.35	0.25	1.00	0.00	0.00	0.92857143	0.48888889	0.58253968
97	0.70	0.40	0.35	0.25	0.50	0.40	0.10	0.96428571	0.50000000	0.53571429
98	0.70	0.40	0.35	0.25	0.50	0.30	0.20	0.96428571	0.50000000	0.53571429
99	0.70	0.40	0.35	0.25	0.60	0.20	0.20	0.96428571	0.48809524	0.54761905
100	0.70	0.40	0.35	0.25	0.40	0.40	0.20	0.96428571	0.50000000	0.53571429

Tabela B.19: Calibração da Instância 7: JHotDraw

n°	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.22531736	0.35314465	1.42153799
2	0.90	0.22488095	0.41612021	1.35899884
3	0.80	0.21794039	0.51250000	1.26955961
4	0.70	0.20583213	0.62907407	1.16509380

Tabela B.20: Calibração da Instância 7: Tomcat

n°	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.73109950	0.43516260	0.83373790
2	0.90	0.68209432	0.56064814	0.75725754
3	0.80	0.66589062	0.61942257	0.71468681
4	0.70	0.66589062	0.69733796	0.63677142

Tabela B.21: Calibração da Instância 7: HSQldb

n°	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.86904762	0.44047619	0.69047619
2	0.90	0.83333333	0.49019608	0.67647059
3	0.80	0.83333333	0.70588235	0.46078431
4	0.70	0.83333333	0.92500000	0.24166667

Tabela B.22: Calibração da Instância 8: JHotDraw

n°	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.27278911	0.60336817	1.12384272
2	0.90	0.27278911	0.61556329	1.11164760
3	0.80	0.26683673	0.72010582	1.01305745
4	0.70	0.26683673	0.83839918	0.89476409

Tabela B.23: Calibração da Instância 8: Tomcat

n°	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.71255611	0.62549603	0.66194785
2	0.90	0.71255611	0.65228174	0.63516213
3	0.80	0.71255611	0.77777777	0.50966610
4	0.70	0.71255611	0.84608378	0.44136009

Tabela B.24: Calibração da Instância 8: HSQldb

n°	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.83333333	0.48148148	0.68518519
2	0.90	0.83333333	0.53703704	0.62962963
3	0.80	0.83333333	0.84210526	0.32456140
4	0.70	0.83333333	0.89473684	0.27192982

Tabela B.25: Calibração da Instância 9: JHotDraw

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.85714285	0.38888888	0.75396827
2	0.90	0.85714285	0.44230769	0.70054946
3	0.80	0.85714285	0.65555555	0.48730160
4	0.70	0.85714285	0.80000000	0.34285715

Tabela B.26: Calibração da Instância 9: Tomcat

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.97222222	0.44374999	0.58402777
2	0.90	0.93055555	0.70357142	0.36587301
3	0.80	0.91666666	0.79947916	0.28385416
4	0.70	0.91666666	0.90990990	0.17342342

Tabela B.27: Calibração da Instância 9: HSQLDB

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.90476190	0.44230769	0.65293040
2	0.90	0.86904762	0.47916667	0.65178571
3	0.80	0.86904762	0.68750000	0.44345238
4	0.70	0.86904762	0.94736842	0.18358396

Tabela B.28: Calibração da Instância 10: JHotDraw

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.24155844	0.76923076	0.98921080
2	0.90	0.23690476	0.76212121	1.00097403
3	0.80	0.21818977	0.74033816	1.04147207
4	0.70	0.20724579	0.78048780	1.01226641

Tabela B.29: Calibração da Instância 10: Tomcat

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.74558080	0.71846282	0.53595636
2	0.90	0.74098875	0.69130434	0.56770689
3	0.80	0.73088482	0.73970037	0.52941479
4	0.70	0.72373015	0.80906148	0.46720835

Tabela B.30: Calibração da Instância 10: HSQLDB

n <sup>o</sup>	<i>distMin</i>	Dispersão	Diversificação	Diferença
1	1.00	0.89285714	0.67777778	0.42936508
2	0.90	0.85714286	0.69607843	0.44677871
3	0.80	0.85714286	0.75000000	0.39285714
4	0.70	0.85714286	0.83333333	0.30952381

Tabela B.31: Calibração da Instância 11: JHotDraw

n <sup>o</sup>	<i>distMin</i>	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
1	1.00	1.00	0.00	0.00	0.92857142	0.25666666	0.81476192
2	1.00	0.50	0.40	0.10	1.00000000	0.12500000	0.87500000
3	1.00	0.50	0.30	0.20	1.00000000	0.14545454	0.85454546
4	1.00	0.60	0.20	0.20	0.92857142	0.16071428	0.91071430
5	1.00	0.40	0.40	0.20	1.00000000	0.16666666	0.83333334
6	0.90	1.00	0.00	0.00	0.92857142	0.25666666	0.81476192
7	0.90	0.50	0.40	0.10	1.00000000	0.29666666	0.70333334
8	0.90	0.50	0.30	0.20	1.00000000	0.28333333	0.71666667
9	0.90	0.60	0.20	0.20	0.92857142	0.30158730	0.76984128
10	0.90	0.40	0.40	0.20	1.00000000	0.28571428	0.71428572
11	0.80	1.00	0.00	0.00	0.92857142	0.40833333	0.66309525
12	0.80	0.50	0.40	0.10	0.92857142	0.33809523	0.73333335
13	0.80	0.50	0.30	0.20	1.00000000	0.30555555	0.69444445
14	0.80	0.60	0.20	0.20	0.92857142	0.29523809	0.77619049
15	0.80	0.40	0.40	0.20	1.00000000	0.35714285	0.64285715
16	0.70	1.00	0.00	0.00	0.92857142	0.44444444	0.62698414
17	0.70	0.50	0.40	0.10	0.92857142	0.40833333	0.66309525
18	0.70	0.50	0.30	0.20	1.00000000	0.41666666	0.58333334
19	0.70	0.60	0.20	0.20	0.92857142	0.46153846	0.60989012
20	0.70	0.40	0.40	0.20	0.92857142	0.40833333	0.66309525

Tabela B.32: Calibração da Instância 11: Tomcat

n <sup>o</sup>	<i>distMin</i>	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
1	1.00	1.00	0.00	0.00	1.00000000	0.18645124	0.81354875
2	1.00	0.50	0.40	0.10	0.95833333	0.04880952	0.99285714
3	1.00	0.50	0.30	0.20	1.00000000	0.05397727	0.94602272
4	1.00	0.60	0.20	0.20	1.00000000	0.06785714	0.93214285
5	1.00	0.40	0.40	0.20	0.97222222	0.05952380	0.96825396
6	0.90	1.00	0.00	0.00	1.00000000	0.18645124	0.81354875
7	0.90	0.50	0.40	0.10	0.94444444	0.11817460	0.93738095
8	0.90	0.50	0.30	0.20	1.00000000	0.16317460	0.83682539
9	0.90	0.60	0.20	0.20	1.00000000	0.13675324	0.86324675
10	0.90	0.40	0.40	0.20	0.94907407	0.13555555	0.91537037
11	0.80	1.00	0.00	0.00	1.00000000	0.36944444	0.63055555
12	0.80	0.50	0.40	0.10	0.93703703	0.32124183	0.74172113
13	0.80	0.50	0.30	0.20	1.00000000	0.35131535	0.64868464
14	0.80	0.60	0.20	0.20	1.00000000	0.35962962	0.64037037

Continua na próxima página ...







... continuação da página anterior										
n <sup>o</sup>	distMin	F <sub>S</sub>	F <sub>O</sub>	F <sub>N</sub>	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
44	0.90	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.38947368	0.61052632
45	0.90	0.35	0.35	0.30	0.40	0.40	0.20	0.98611111	0.41811594	0.59577295
46	0.90	0.40	0.35	0.25	1.00	0.00	0.00	1.00000000	0.44466667	0.55533333
47	0.90	0.40	0.35	0.25	0.50	0.40	0.10	0.98611111	0.42847222	0.58541667
48	0.90	0.40	0.35	0.25	0.50	0.30	0.20	0.98611111	0.43541667	0.57847222
49	0.90	0.40	0.35	0.25	0.60	0.20	0.20	0.97222222	0.42500000	0.60277778
50	0.90	0.40	0.35	0.25	0.40	0.40	0.20	0.98611111	0.45800000	0.55588889
51	0.80	0.40	0.40	0.20	1.00	0.00	0.00	0.97222222	0.72222222	0.30555556
52	0.80	0.40	0.40	0.20	0.50	0.40	0.10	0.95370370	0.69444444	0.35185185
53	0.80	0.40	0.40	0.20	0.50	0.30	0.20	0.95370370	0.65885417	0.38744213
54	0.80	0.40	0.40	0.20	0.60	0.20	0.20	0.95833333	0.70572917	0.33593750
55	0.80	0.40	0.40	0.20	0.40	0.40	0.20	0.91203704	0.71064815	0.37731481
56	0.80	0.50	0.30	0.20	1.00	0.00	0.00	0.97222222	0.72656250	0.30121528
57	0.80	0.50	0.30	0.20	0.50	0.40	0.10	0.95833333	0.71388889	0.32777778
58	0.80	0.50	0.30	0.20	0.50	0.30	0.20	0.95833333	0.68103448	0.36063218
59	0.80	0.50	0.30	0.20	0.60	0.20	0.20	0.95833333	0.66954023	0.37212644
60	0.80	0.50	0.30	0.20	0.40	0.40	0.20	0.94444444	0.68103448	0.37452107
61	0.80	0.45	0.45	0.10	1.00	0.00	0.00	0.93750000	0.80341880	0.25908120
62	0.80	0.45	0.45	0.10	0.50	0.40	0.10	0.93750000	0.76754386	0.29495614
63	0.80	0.45	0.45	0.10	0.50	0.30	0.20	0.93750000	0.76754386	0.29495614
64	0.80	0.45	0.45	0.10	0.60	0.20	0.20	0.93750000	0.77777778	0.28472222
65	0.80	0.45	0.45	0.10	0.40	0.40	0.20	0.90972222	0.76068376	0.32959402
66	0.80	0.35	0.35	0.30	1.00	0.00	0.00	0.97222222	0.52586207	0.50191571
67	0.80	0.35	0.35	0.30	0.50	0.40	0.10	0.95833333	0.60057471	0.44109195
68	0.80	0.35	0.35	0.30	0.50	0.30	0.20	0.97222222	0.58333333	0.44444444
69	0.80	0.35	0.35	0.30	0.60	0.20	0.20	0.97222222	0.51436782	0.51340996
70	0.80	0.35	0.35	0.30	0.40	0.40	0.20	0.95833333	0.56944444	0.47222222
71	0.80	0.40	0.35	0.25	1.00	0.00	0.00	0.97222222	0.52586207	0.50191571
72	0.80	0.40	0.35	0.25	0.50	0.40	0.10	0.97222222	0.66379310	0.36398467
73	0.80	0.40	0.35	0.25	0.50	0.30	0.20	0.97222222	0.63505747	0.39272031
74	0.80	0.40	0.35	0.25	0.60	0.20	0.20	0.97222222	0.61206897	0.41570881
75	0.80	0.40	0.35	0.25	0.40	0.40	0.20	0.97222222	0.65277778	0.37500000
76	0.70	0.40	0.40	0.20	1.00	0.00	0.00	0.95833333	0.90789474	0.13377193
77	0.70	0.40	0.40	0.20	0.50	0.40	0.10	0.93750000	0.87083333	0.19166667
78	0.70	0.40	0.40	0.20	0.50	0.30	0.20	0.93750000	0.87500000	0.18750000
79	0.70	0.40	0.40	0.20	0.60	0.20	0.20	0.95833333	0.90540541	0.13626126
80	0.70	0.40	0.40	0.20	0.40	0.40	0.20	0.90972222	0.86585366	0.22442412
81	0.70	0.50	0.30	0.20	1.00	0.00	0.00	0.95833333	0.91441441	0.12725225
82	0.70	0.50	0.30	0.20	0.50	0.40	0.10	0.94444444	0.89639640	0.15915916
83	0.70	0.50	0.30	0.20	0.50	0.30	0.20	0.95833333	0.91441441	0.12725225
84	0.70	0.50	0.30	0.20	0.60	0.20	0.20	0.95833333	0.90090090	0.14076577
85	0.70	0.50	0.30	0.20	0.40	0.40	0.20	0.94444444	0.87962963	0.17592593
86	0.70	0.45	0.45	0.10	1.00	0.00	0.00	0.93750000	0.89024390	0.17225610
87	0.70	0.45	0.45	0.10	0.50	0.40	0.10	0.90277778	0.90000000	0.19722222
88	0.70	0.45	0.45	0.10	0.50	0.30	0.20	0.93287037	0.89285714	0.17427249
89	0.70	0.45	0.45	0.10	0.60	0.20	0.20	0.93750000	0.89024390	0.17225610
90	0.70	0.45	0.45	0.10	0.40	0.40	0.20	0.89351852	0.89130435	0.21517713
91	0.70	0.35	0.35	0.30	1.00	0.00	0.00	0.97222222	0.81481481	0.21296296
92	0.70	0.35	0.35	0.30	0.50	0.40	0.10	0.93055556	0.80714286	0.26230159
93	0.70	0.35	0.35	0.30	0.50	0.30	0.20	0.95833333	0.85648148	0.18518519
94	0.70	0.35	0.35	0.30	0.60	0.20	0.20	0.95833333	0.88425926	0.15740741
95	0.70	0.35	0.35	0.30	0.40	0.40	0.20	0.93055556	0.80092593	0.26851852
96	0.70	0.40	0.35	0.25	1.00	0.00	0.00	0.95833333	0.88738739	0.15427928
97	0.70	0.40	0.35	0.25	0.50	0.40	0.10	0.95833333	0.90090090	0.14076577
98	0.70	0.40	0.35	0.25	0.50	0.30	0.20	0.95833333	0.90090090	0.14076577
99	0.70	0.40	0.35	0.25	0.60	0.20	0.20	0.95833333	0.90090090	0.14076577
100	0.70	0.40	0.35	0.25	0.40	0.40	0.20	0.95833333	0.88738739	0.15427928

Tabela B.36: Calibração da Instância 12: HSQldb

n <sup>o</sup>	distMin	F <sub>S</sub>	F <sub>O</sub>	F <sub>N</sub>	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
1	1.00	0.40	0.40	0.20	1.00	0.00	0.00	0.96428571	0.29000000	0.74571429
2	1.00	0.40	0.40	0.20	0.50	0.40	0.10	1.00000000	0.19166667	0.80833333
3	1.00	0.40	0.40	0.20	0.50	0.30	0.20	1.00000000	0.23750000	0.76250000
4	1.00	0.40	0.40	0.20	0.60	0.20	0.20	1.00000000	0.19166667	0.80833333
5	1.00	0.40	0.40	0.20	0.40	0.40	0.20	1.00000000	0.12698413	0.87301587
6	1.00	0.50	0.30	0.20	1.00	0.00	0.00	0.96428571	0.30833333	0.72738095
7	1.00	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.17777778	0.82222222
8	1.00	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.27083333	0.72916667
9	1.00	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.28030303	0.71969697
10	1.00	0.50	0.30	0.20	0.40	0.40	0.20	1.00000000	0.20833333	0.79166667
11	1.00	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.30000000	0.73571429
12	1.00	0.45	0.45	0.10	0.50	0.40	0.10	0.96428571	0.35277778	0.68293651
13	1.00	0.45	0.45	0.10	0.50	0.30	0.20	0.96428571	0.23154762	0.80416667
14	1.00	0.45	0.45	0.10	0.60	0.20	0.20	1.00000000	0.30666667	0.69333333
15	1.00	0.45	0.45	0.10	0.40	0.40	0.20	0.96428571	0.33166667	0.70404762
16	1.00	0.35	0.35	0.30	1.00	0.00	0.00	1.00000000	0.31666667	0.68333333
17	1.00	0.35	0.35	0.30	0.50	0.40	0.10	1.00000000	0.22777778	0.77222222
18	1.00	0.35	0.35	0.30	0.50	0.30	0.20	1.00000000	0.23750000	0.76250000
19	1.00	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.19166667	0.80833333
20	1.00	0.35	0.35	0.30	0.40	0.40	0.20	1.00000000	0.18650794	0.81349206
21	1.00	0.40	0.35	0.25	1.00	0.00	0.00	0.96428571	0.29000000	0.74571429
22	1.00	0.40	0.35	0.25	0.50	0.40	0.10	1.00000000	0.26666667	0.73333333
23	1.00	0.40	0.35	0.25	0.50	0.30	0.20	1.00000000	0.28958333	0.71041667
24	1.00	0.40	0.35	0.25	0.60	0.20	0.20	1.00000000	0.18650794	0.81349206
25	1.00	0.40	0.35	0.25	0.40	0.40	0.20	1.00000000	0.23333333	0.76666667
26	0.90	0.40	0.40	0.20	1.00	0.00	0.00	0.96428571	0.48717949	0.54853480
27	0.90	0.40	0.40	0.20	0.50	0.40	0.10	0.96428571	0.44696970	0.58874459
28	0.90	0.40	0.40	0.20	0.50	0.30	0.20	0.96428571	0.42500000	0.61071429
29	0.90	0.40	0.40	0.20	0.60	0.20	0.20	0.96428571	0.43939394	0.59632035
30	0.90	0.40	0.40	0.20	0.40	0.40	0.20	0.96428571	0.37500000	0.66071429

Continua na próxima página ...

... continuação da página anterior										
n <sup>o</sup>	distMin	F <sub>S</sub>	F <sub>O</sub>	F <sub>N</sub>	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
31	0.90	0.50	0.30	0.20	1.00	0.00	0.00	0.96428571	0.46527778	0.57043651
32	0.90	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.46212121	0.53787879
33	0.90	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.44166667	0.55833333
34	0.90	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.45454545	0.54545455
35	0.90	0.50	0.30	0.20	0.40	0.40	0.20	1.00000000	0.39583333	0.60416667
36	0.90	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.51388889	0.52182540
37	0.90	0.45	0.45	0.10	0.50	0.40	0.10	0.96428571	0.47435897	0.56135531
38	0.90	0.45	0.45	0.10	0.50	0.30	0.20	0.96428571	0.47435897	0.56135531
39	0.90	0.45	0.45	0.10	0.60	0.20	0.20	0.96428571	0.47435897	0.56135531
40	0.90	0.45	0.45	0.10	0.40	0.40	0.20	0.96428571	0.47435897	0.56135531
41	0.90	0.35	0.35	0.30	1.00	0.00	0.00	0.96428571	0.45757576	0.57813853
42	0.90	0.35	0.35	0.30	0.50	0.40	0.10	1.00000000	0.42500000	0.57500000
43	0.90	0.35	0.35	0.30	0.50	0.30	0.20	1.00000000	0.42962963	0.57037037
44	0.90	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.44444444	0.55555556
45	0.90	0.35	0.35	0.30	0.40	0.40	0.20	1.00000000	0.36190476	0.63809524
46	0.90	0.40	0.35	0.25	1.00	0.00	0.00	0.96428571	0.48076923	0.55494505
47	0.90	0.40	0.35	0.25	0.50	0.40	0.10	1.00000000	0.42962963	0.57037037
48	0.90	0.40	0.35	0.25	0.50	0.30	0.20	1.00000000	0.42962963	0.57037037
49	0.90	0.40	0.35	0.25	0.60	0.20	0.20	0.96428571	0.42500000	0.61071429
50	0.90	0.40	0.35	0.25	0.40	0.40	0.20	1.00000000	0.40000000	0.60000000
51	0.80	0.40	0.40	0.20	1.00	0.00	0.00	0.92857143	0.71875000	0.35267857
52	0.80	0.40	0.40	0.20	0.50	0.40	0.10	0.92857143	0.56666667	0.50476190
53	0.80	0.40	0.40	0.20	0.50	0.30	0.20	0.92857143	0.60000000	0.47142857
54	0.80	0.40	0.40	0.20	0.60	0.20	0.20	0.92857143	0.63333333	0.43809524
55	0.80	0.40	0.40	0.20	0.40	0.40	0.20	0.92857143	0.56666667	0.50476190
56	0.80	0.50	0.30	0.20	1.00	0.00	0.00	0.96428571	0.73333333	0.30238095
57	0.80	0.50	0.30	0.20	0.50	0.40	0.10	0.96428571	0.53571429	0.50000000
58	0.80	0.50	0.30	0.20	0.50	0.30	0.20	0.96428571	0.53571429	0.50000000
59	0.80	0.50	0.30	0.20	0.60	0.20	0.20	0.96428571	0.57142857	0.46428571
60	0.80	0.50	0.30	0.20	0.40	0.40	0.20	0.96428571	0.53571429	0.50000000
61	0.80	0.45	0.45	0.10	1.00	0.00	0.00	0.92857143	0.71875000	0.35267857
62	0.80	0.45	0.45	0.10	0.50	0.40	0.10	0.92857143	0.68750000	0.38392857
63	0.80	0.45	0.45	0.10	0.50	0.30	0.20	0.92857143	0.68750000	0.38392857
64	0.80	0.45	0.45	0.10	0.60	0.20	0.20	0.92857143	0.71875000	0.35267857
65	0.80	0.45	0.45	0.10	0.40	0.40	0.20	0.92857143	0.68750000	0.38392857
66	0.80	0.35	0.35	0.30	1.00	0.00	0.00	0.96428571	0.56410256	0.47161172
67	0.80	0.35	0.35	0.30	0.50	0.40	0.10	0.96428571	0.55128205	0.48443223
68	0.80	0.35	0.35	0.30	0.50	0.30	0.20	0.96428571	0.56410256	0.47161172
69	0.80	0.35	0.35	0.30	0.60	0.20	0.20	0.96428571	0.55952381	0.47619048
70	0.80	0.35	0.35	0.30	0.40	0.40	0.20	0.96428571	0.55128205	0.48443223
71	0.80	0.40	0.35	0.25	1.00	0.00	0.00	0.96428571	0.60256410	0.43315018
72	0.80	0.40	0.35	0.25	0.50	0.40	0.10	0.96428571	0.56410256	0.47161172
73	0.80	0.40	0.35	0.25	0.50	0.30	0.20	0.96428571	0.57142857	0.46428571
74	0.80	0.40	0.35	0.25	0.60	0.20	0.20	0.96428571	0.60714286	0.42857143
75	0.80	0.40	0.35	0.25	0.40	0.40	0.20	0.96428571	0.51282051	0.52289377
76	0.70	0.40	0.40	0.20	1.00	0.00	0.00	0.92857143	0.84375000	0.22767857
77	0.70	0.40	0.40	0.20	0.50	0.40	0.10	0.92857143	0.87500000	0.19642857
78	0.70	0.40	0.40	0.20	0.50	0.30	0.20	0.92857143	0.87500000	0.19642857
79	0.70	0.40	0.40	0.20	0.60	0.20	0.20	0.92857143	0.87500000	0.19642857
80	0.70	0.40	0.40	0.20	0.40	0.40	0.20	0.92857143	0.84375000	0.22767857
81	0.70	0.50	0.30	0.20	1.00	0.00	0.00	0.96428571	0.83333333	0.20238095
82	0.70	0.50	0.30	0.20	0.50	0.40	0.10	0.96428571	0.83333333	0.20238095
83	0.70	0.50	0.30	0.20	0.50	0.30	0.20	0.96428571	0.83333333	0.20238095
84	0.70	0.50	0.30	0.20	0.60	0.20	0.20	0.96428571	0.86666667	0.16904762
85	0.70	0.50	0.30	0.20	0.40	0.40	0.20	0.96428571	0.86666667	0.16904762
86	0.70	0.45	0.45	0.10	1.00	0.00	0.00	0.92857143	0.87500000	0.19642857
87	0.70	0.45	0.45	0.10	0.50	0.40	0.10	0.92857143	0.87500000	0.19642857
88	0.70	0.45	0.45	0.10	0.50	0.30	0.20	0.92857143	0.87500000	0.19642857
89	0.70	0.45	0.45	0.10	0.60	0.20	0.20	0.92857143	0.87500000	0.19642857
90	0.70	0.45	0.45	0.10	0.40	0.40	0.20	0.92857143	0.87500000	0.19642857
91	0.70	0.35	0.35	0.30	1.00	0.00	0.00	0.96428571	0.73333333	0.30238095
92	0.70	0.35	0.35	0.30	0.50	0.40	0.10	0.96428571	0.76666667	0.26904762
93	0.70	0.35	0.35	0.30	0.50	0.30	0.20	0.96428571	0.73333333	0.30238095
94	0.70	0.35	0.35	0.30	0.60	0.20	0.20	0.96428571	0.70000000	0.33571429
95	0.70	0.35	0.35	0.30	0.40	0.40	0.20	0.96428571	0.63333333	0.40238095
96	0.70	0.40	0.35	0.25	1.00	0.00	0.00	0.96428571	0.83333333	0.20238095
97	0.70	0.40	0.35	0.25	0.50	0.40	0.10	0.96428571	0.83333333	0.20238095
98	0.70	0.40	0.35	0.25	0.50	0.30	0.20	0.96428571	0.83333333	0.20238095
99	0.70	0.40	0.35	0.25	0.60	0.20	0.20	0.96428571	0.83333333	0.20238095
100	0.70	0.40	0.35	0.25	0.40	0.40	0.20	0.96428571	0.83333333	0.20238095

Tabela B.37: Calibração da Instância 13: JHotDraw

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.19649629	0.65646258	1.14704113
2	4	1.0	0.24397675	0.41716666	1.33885659
3	5	1.0	0.22399749	0.39477124	1.38123127
4	3	0.8	0.21098550	0.65217391	1.13684059
5	4	0.8	0.22788012	0.41200000	1.36011988
6	5	0.8	0.24238945	0.38940972	1.36820083
7	3	1.2	0.19640180	0.65833333	1.14526487
8	4	1.2	0.23568028	0.45570175	1.30861797
9	5	1.2	0.22096088	0.41994047	1.35909865

Tabela B.38: Calibração da Instância 13: Tomcat

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.70879072	0.73108108	0.56012819
2	4	1.0	0.79891513	0.45347008	0.74761478
3	5	1.0	0.75021549	0.41308506	0.83669944

Continua na próxima página ...

... continuação da página anterior

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
4	3	0.8	0.71857879	0.75096801	0.53045318
5	4	0.8	0.79668384	0.45022831	0.75308784
6	5	0.8	0.77801293	0.39108843	0.83089863
7	3	1.2	0.70875597	0.73999999	0.55124402
8	4	1.2	0.73234112	0.48557585	0.78208301
9	5	1.2	0.74190182	0.40802005	0.85007812

Tabela B.39: Calibração da Instância 13: HSQLDB

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.84523810	0.72222222	0.43253968
2	4	1.0	0.84523810	0.56666667	0.58809524
3	5	1.0	0.84523810	0.49621212	0.65854978
4	3	0.8	0.84523810	0.69607843	0.45868347
5	4	0.8	0.84523810	0.55357143	0.60119048
6	5	0.8	0.85714286	0.47222222	0.67063492
7	3	1.2	0.84523810	0.72222222	0.43253968
8	4	1.2	0.84523810	0.59761905	0.55714286
9	5	1.2	0.88095238	0.55681818	0.56222944

Tabela B.40: Calibração da Instância 14: JHotDraw

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.19611935	0.52450980	1.27937085
2	4	1.0	0.20181306	0.48878695	1.30939999
3	5	1.0	0.22011904	0.43624497	1.34363599
4	3	0.8	0.19888451	0.52592592	1.03782911
5	4	0.8	0.24552036	0.48649021	1.26798943
6	5	0.8	0.24630230	0.46265581	1.29104189
7	3	1.2	0.19835600	0.52479338	1.27685062
8	4	1.2	0.20797430	0.48598820	1.30603750
9	5	1.2	0.22037509	0.46124482	1.31838009

Tabela B.41: Calibração da Instância 14: Tomcat

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.69410658	0.51434878	0.79154462
2	4	1.0	0.72707155	0.46701388	0.80591455
3	5	1.0	0.70890999	0.43850611	0.85258389
4	3	0.8	0.69468000	0.51923937	0.78608062
5	4	0.8	0.71782147	0.47281576	0.80936276
6	5	0.8	0.71242503	0.45358205	0.83399290
7	3	1.2	0.69427127	0.51666666	0.78906205
8	4	1.2	0.71311991	0.46842447	0.81845560
9	5	1.2	0.70914049	0.43202697	0.85883253

Tabela B.42: Calibração da Instância 14: HSQLDB

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.83333333	0.77500000	0.39166667
2	4	1.0	0.83333333	0.61764706	0.54901961
3	5	1.0	0.83333333	0.52380952	0.64285714
4	3	0.8	0.83333333	0.70175439	0.46491228
5	4	0.8	0.83333333	0.60544218	0.56122449
6	5	0.8	0.83333333	0.52380952	0.64285714
7	3	1.2	0.83333333	0.77500000	0.39166667
8	4	1.2	0.83333333	0.61764706	0.54901961
9	5	1.2	0.83333333	0.55208333	0.61458333

Tabela B.43: Calibração da Instância 15: JHotDraw

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.74829931	0.88709677	0.36460392
2	4	1.0	0.81190476	0.82954545	0.35854979
3	5	1.0	0.81785714	0.75833333	0.42380953
4	3	0.8	0.83571428	0.80000000	0.36428572
5	4	0.8	0.83928571	0.76315789	0.39755640
6	5	0.8	0.83571428	0.72807017	0.43621555
7	3	1.2	0.70238095	0.93750000	0.36011905
8	4	1.2	0.80952380	0.82000000	0.37047620
9	5	1.2	0.81190476	0.77536231	0.41273293

Tabela B.44: Calibração da Instância 15: Tomcat

n <sup>o</sup>	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.85771604	0.87356321	0.26872073
2	4	1.0	0.87060185	0.82924528	0.30015286
3	5	1.0	0.89494047	0.79528985	0.30976966
4	3	0.8	0.88082010	0.85034013	0.26883975
5	4	0.8	0.87619047	0.80326086	0.32054865
6	5	0.8	0.88082010	0.79734848	0.32183140
7	3	1.2	0.85231481	0.87566137	0.27202380
8	4	1.2	0.86944444	0.84454545	0.28601010
9	5	1.2	0.87549603	0.80782312	0.31668083

Tabela B.45: Calibração da Instância 15: HSQldb

n°	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.86904762	0.72549020	0.40546218
2	4	1.0	0.89285714	0.50333333	0.60380952
3	5	1.0	0.89285714	0.38244048	0.72470238
4	3	0.8	0.86904762	0.61111111	0.51984127
5	4	0.8	0.89285714	0.50333333	0.60380952
6	5	0.8	0.89285714	0.48611111	0.62103175
7	3	1.2	0.86904762	0.72549020	0.40546218
8	4	1.2	0.86904762	0.58333333	0.54761905
9	5	1.2	0.89285714	0.53472222	0.57242063

Tabela B.46: Calibração da Instância 16: JHotDraw

n°	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.22062074	0.57352941	1.20584985
2	4	1.0	0.23472222	0.53954802	1.22572976
3	5	1.0	0.24047619	0.46797385	1.29154996
4	3	0.8	0.22393707	0.58333333	1.19272960
5	4	0.8	0.24889455	0.51722391	1.23388154
6	5	0.8	0.25119047	0.47839506	1.27041447
7	3	1.2	0.20133477	0.62359550	1.17506973
8	4	1.2	0.23222789	0.53672316	1.23104895
9	5	1.2	0.23888888	0.47450980	1.28660132

Tabela B.47: Calibração da Instância 16: Tomcat

n°	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.72279342	0.64242424	0.63478233
2	4	1.0	0.74770923	0.60912087	0.64316988
3	5	1.0	0.75032066	0.54889023	0.70078909
4	3	0.8	0.72355193	0.62943262	0.64701544
5	4	0.8	0.74219777	0.60553221	0.65227001
6	5	0.8	0.76697530	0.55733082	0.67569386
7	3	1.2	0.71987753	0.65086206	0.62926039
8	4	1.2	0.74555776	0.63178053	0.62266170
9	5	1.2	0.74843304	0.55404040	0.69752654

Tabela B.48: Calibração da Instância 16: HSQldb

n°	tam_part	sim_min	Dispersão	Diversificação	Diferença
1	3	1.0	0.89285714	0.51428571	0.59285714
2	4	1.0	0.92857143	0.47692308	0.59450549
3	5	1.0	0.89285714	0.46410256	0.64304029
4	3	0.8	0.89285714	0.51428571	0.59285714
5	4	0.8	0.92857143	0.48214286	0.58928571
6	5	0.8	0.96428571	0.47500000	0.56071429
7	3	1.2	0.85714286	0.57843137	0.56442577
8	4	1.2	0.92857143	0.47692308	0.59450549
9	5	1.2	0.89285714	0.46410256	0.64304029

Tabela B.49: Calibração da Instância 17: JHotDraw

n°	tam_part	sim_min	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
1	3	1.0	1.00	0.00	0.00	0.67627551	0.75225225	0.57147224
2	3	1.0	0.50	0.40	0.10	0.84523809	0.76811594	0.38664597
3	3	1.0	0.50	0.30	0.20	0.88163265	0.74206349	0.37630386
4	3	1.0	0.60	0.20	0.20	0.88571428	0.75396825	0.36031747
5	3	1.0	0.40	0.40	0.20	0.84523809	0.78472222	0.37003969
6	4	1.0	1.00	0.00	0.00	0.69115646	0.72916666	0.57967688
7	4	1.0	0.50	0.40	0.10	0.87925170	0.72727272	0.39347558
8	4	1.0	0.50	0.30	0.20	0.84523809	0.68115942	0.47360249
9	4	1.0	0.60	0.20	0.20	0.87142857	0.67543859	0.45313284
10	4	1.0	0.40	0.40	0.20	0.88095238	0.73484848	0.38419914
11	5	1.0	1.00	0.00	0.00	0.69166666	0.71111111	0.59722223
12	5	1.0	0.50	0.40	0.10	0.88095238	0.71212121	0.40692641
13	5	1.0	0.50	0.30	0.20	0.84523809	0.72222222	0.43253969
14	5	1.0	0.60	0.20	0.20	0.90476190	0.71078431	0.38445379
15	5	1.0	0.40	0.40	0.20	0.88571428	0.67647058	0.43781514
16	3	0.8	1.00	0.00	0.00	0.68690476	0.73737373	0.57572151
17	3	0.8	0.50	0.40	0.10	0.88095238	0.74206349	0.37698413
18	3	0.8	0.50	0.30	0.20	0.88333333	0.71228070	0.40438597
19	3	0.8	0.60	0.20	0.20	0.88571428	0.73015873	0.38412699
20	3	0.8	0.40	0.40	0.20	0.88571428	0.73333333	0.38095239
21	4	0.8	1.00	0.00	0.00	0.74642857	0.69753086	0.55604057
22	4	0.8	0.50	0.40	0.10	0.88095238	0.71428571	0.40476191
23	4	0.8	0.50	0.30	0.20	0.90476190	0.63095238	0.46428572
24	4	0.8	0.60	0.20	0.20	0.90714285	0.68627450	0.40658265
25	4	0.8	0.40	0.40	0.20	0.88571428	0.68421052	0.43007520
26	5	0.8	1.00	0.00	0.00	0.77380952	0.65151515	0.57467533
27	5	0.8	0.50	0.40	0.10	0.88571428	0.68981481	0.42447091
28	5	0.8	0.50	0.30	0.20	0.91071428	0.56944444	0.51984128
29	5	0.8	0.60	0.20	0.20	0.91071428	0.68846153	0.40082419
30	5	0.8	0.40	0.40	0.20	0.89285714	0.63333333	0.47380953
31	3	1.2	1.00	0.00	0.00	0.67043650	0.80833333	0.52123017
32	3	1.2	0.50	0.40	0.10	0.82539682	0.84946236	0.32514082
33	3	1.2	0.50	0.30	0.20	0.80257936	0.81111111	0.38630953
34	3	1.2	0.60	0.20	0.20	0.87755102	0.81333333	0.30911565
35	3	1.2	0.40	0.40	0.20	0.79421768	0.76282051	0.44296181
36	4	1.2	1.00	0.00	0.00	0.68112244	0.78095238	0.53792518
37	4	1.2	0.50	0.40	0.10	0.87500000	0.79487179	0.33012821

Continua na próxima página ...

... continuação da página anterior								
n°	<i>tam_part</i>	<i>sim_min</i>	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
38	4	1.2	0.50	0.30	0.20	0.83928571	0.76543209	0.39528220
39	4	1.2	0.60	0.20	0.20	0.90306122	0.75462962	0.34230916
40	4	1.2	0.40	0.40	0.20	0.84183673	0.72222222	0.43594105
41	5	1.2	1.00	0.00	0.00	0.68928571	0.72043010	0.59028419
42	5	1.2	0.50	0.40	0.10	0.84523809	0.76811594	0.38664597
43	5	1.2	0.50	0.30	0.20	0.88095238	0.73484848	0.38419914
44	5	1.2	0.60	0.20	0.20	0.88095238	0.78030303	0.33874459
45	5	1.2	0.40	0.40	0.20	0.84523809	0.71969696	0.43506495

Tabela B.50: Calibração da Instância 17: Tomcat

n°	<i>tam_part</i>	<i>sim_min</i>	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
1	3	1.0	1.00	0.00	0.00	0.88324175	0.75264550	0.36411273
2	3	1.0	0.50	0.40	0.10	0.88811728	0.63219696	0.47968574
3	3	1.0	0.50	0.30	0.20	0.91481481	0.67037037	0.41481481
4	3	1.0	0.60	0.20	0.20	0.91250000	0.66998892	0.41751107
5	3	1.0	0.40	0.40	0.20	0.88518518	0.64743589	0.46737891
6	4	1.0	1.00	0.00	0.00	0.88703703	0.70601851	0.40694444
7	4	1.0	0.50	0.40	0.10	0.91527777	0.62027914	0.46444307
8	4	1.0	0.50	0.30	0.20	0.94027777	0.66436781	0.39535440
9	4	1.0	0.60	0.20	0.20	0.90879629	0.70097402	0.39022967
10	4	1.0	0.40	0.40	0.20	0.88016975	0.64246031	0.47736992
11	5	1.0	1.00	0.00	0.00	0.89488636	0.63136686	0.47374677
12	5	1.0	0.50	0.40	0.10	0.90817901	0.54259259	0.54922839
13	5	1.0	0.50	0.30	0.20	0.93055555	0.54531250	0.52413194
14	5	1.0	0.60	0.20	0.20	0.91759259	0.65174603	0.43066138
15	5	1.0	0.40	0.40	0.20	0.91064814	0.55208333	0.53726851
16	3	0.8	1.00	0.00	0.00	0.88817340	0.74640610	0.36542048
17	3	0.8	0.50	0.40	0.10	0.92685185	0.58281250	0.49033564
18	3	0.8	0.50	0.30	0.20	0.94058641	0.64175627	0.41765730
19	3	0.8	0.60	0.20	0.20	0.91512345	0.72254901	0.36232752
20	3	0.8	0.40	0.40	0.20	0.90231481	0.61666666	0.48101851
21	4	0.8	1.00	0.00	0.00	0.89930555	0.68645833	0.41423611
22	4	0.8	0.50	0.40	0.10	0.92592592	0.62360446	0.45046960
23	4	0.8	0.50	0.30	0.20	0.94058641	0.67037037	0.38904320
24	4	0.8	0.60	0.20	0.20	0.91512345	0.73030303	0.35457351
25	4	0.8	0.40	0.40	0.20	0.89999999	0.61523809	0.48476190
26	5	0.8	1.00	0.00	0.00	0.91898148	0.61786987	0.46314864
27	5	0.8	0.50	0.40	0.10	0.91759259	0.56296296	0.51944444
28	5	0.8	0.50	0.30	0.20	0.94675925	0.56388888	0.48935185
29	5	0.8	0.60	0.20	0.20	0.96296296	0.53650793	0.50052910
30	5	0.8	0.40	0.40	0.20	0.92916666	0.53371428	0.53711904
31	3	1.2	1.00	0.00	0.00	0.87999338	0.75821596	0.36179065
32	3	1.2	0.50	0.40	0.10	0.86255611	0.69406779	0.44337608
33	3	1.2	0.50	0.30	0.20	0.88317901	0.69607843	0.42074255
34	3	1.2	0.60	0.20	0.20	0.88564814	0.71058201	0.40376984
35	3	1.2	0.40	0.40	0.20	0.86710858	0.61666666	0.51622474
36	4	1.2	1.00	0.00	0.00	0.88842592	0.75000000	0.36157407
37	4	1.2	0.50	0.40	0.10	0.88657407	0.63122448	0.48220143
38	4	1.2	0.50	0.30	0.20	0.88317901	0.69097222	0.42584876
39	4	1.2	0.60	0.20	0.20	0.89583333	0.71233766	0.39182900
40	4	1.2	0.40	0.40	0.20	0.86353615	0.66550387	0.47095996
41	5	1.2	1.00	0.00	0.00	0.89189814	0.69631410	0.41178774
42	5	1.2	0.50	0.40	0.10	0.88089726	0.62102272	0.49808000
43	5	1.2	0.50	0.30	0.20	0.91137566	0.57675438	0.51186995
44	5	1.2	0.60	0.20	0.20	0.93452380	0.61802884	0.44744734
45	5	1.2	0.40	0.40	0.20	0.89976851	0.56495726	0.53527421

Tabela B.51: Calibração da Instância 17: HSQLDB

n°	<i>tam_part</i>	<i>sim_min</i>	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
1	3	1.0	1.00	0.00	0.00	1.00000000	0.56060606	0.43939394
2	3	1.0	0.50	0.40	0.10	1.00000000	0.50757576	0.49242424
3	3	1.0	0.50	0.30	0.20	1.00000000	0.43452381	0.56547619
4	3	1.0	0.60	0.20	0.20	1.00000000	0.51587302	0.48412698
5	3	1.0	0.40	0.40	0.20	1.00000000	0.43888889	0.56111111
6	4	1.0	1.00	0.00	0.00	1.00000000	0.46969697	0.53030303
7	4	1.0	0.50	0.40	0.10	1.00000000	0.50000000	0.50000000
8	4	1.0	0.50	0.30	0.20	1.00000000	0.46212121	0.53787879
9	4	1.0	0.60	0.20	0.20	1.00000000	0.49166667	0.50833333
10	4	1.0	0.40	0.40	0.20	1.00000000	0.46212121	0.53787879
11	5	1.0	1.00	0.00	0.00	1.00000000	0.44444444	0.55555556
12	5	1.0	0.50	0.40	0.10	1.00000000	0.41041667	0.58958333
13	5	1.0	0.50	0.30	0.20	1.00000000	0.44444444	0.55555556
14	5	1.0	0.60	0.20	0.20	1.00000000	0.42708333	0.57291667
15	5	1.0	0.40	0.40	0.20	1.00000000	0.45000000	0.55000000
16	3	0.8	1.00	0.00	0.00	1.00000000	0.55000000	0.45000000
17	3	0.8	0.50	0.40	0.10	1.00000000	0.51587302	0.48412698
18	3	0.8	0.50	0.30	0.20	1.00000000	0.43333333	0.56666667
19	3	0.8	0.60	0.20	0.20	1.00000000	0.45312500	0.54687500
20	3	0.8	0.40	0.40	0.20	1.00000000	0.39047619	0.60952381
21	4	0.8	1.00	0.00	0.00	1.00000000	0.45833333	0.54166667
22	4	0.8	0.50	0.40	0.10	1.00000000	0.43333333	0.56666667
23	4	0.8	0.50	0.30	0.20	1.00000000	0.43333333	0.56666667
24	4	0.8	0.60	0.20	0.20	1.00000000	0.51587302	0.48412698
25	4	0.8	0.40	0.40	0.20	1.00000000	0.39047619	0.60952381
26	5	0.8	1.00	0.00	0.00	1.00000000	0.45312500	0.54687500
27	5	0.8	0.50	0.40	0.10	1.00000000	0.40740741	0.59259259
28	5	0.8	0.50	0.30	0.20	1.00000000	0.43452381	0.56547619
29	5	0.8	0.60	0.20	0.20	1.00000000	0.45312500	0.54687500

Continua na próxima página ...

... continuação da página anterior								
n <sup>o</sup>	tam-part	sim.min	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
30	5	0.8	0.40	0.40	0.20	1.00000000	0.44444444	0.55555556
31	3	1.2	1.00	0.00	0.00	1.00000000	0.56944444	0.43055556
32	3	1.2	0.50	0.40	0.10	1.00000000	0.49444444	0.50555556
33	3	1.2	0.50	0.30	0.20	0.96428571	0.50000000	0.53571429
34	3	1.2	0.60	0.20	0.20	1.00000000	0.56363636	0.43636364
35	3	1.2	0.40	0.40	0.20	0.96428571	0.49242424	0.54329004
36	4	1.2	1.00	0.00	0.00	1.00000000	0.56060606	0.43939394
37	4	1.2	0.50	0.40	0.10	1.00000000	0.50000000	0.50000000
38	4	1.2	0.50	0.30	0.20	1.00000000	0.44444444	0.55555556
39	4	1.2	0.60	0.20	0.20	1.00000000	0.52083333	0.47916667
40	4	1.2	0.40	0.40	0.20	1.00000000	0.46212121	0.53787879
41	5	1.2	1.00	0.00	0.00	1.00000000	0.46666667	0.53333333
42	5	1.2	0.50	0.40	0.10	1.00000000	0.41041667	0.58958333
43	5	1.2	0.50	0.30	0.20	1.00000000	0.44444444	0.55555556
44	5	1.2	0.60	0.20	0.20	1.00000000	0.46212121	0.53787879
45	5	1.2	0.40	0.40	0.20	1.00000000	0.43888889	0.56111111

Tabela B.52: Calibração da Instância 18: JHotDraw

n <sup>o</sup>	tp	smin	F <sub>S</sub>	F <sub>O</sub>	F <sub>N</sub>	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
1	3	1.0	0.40	0.40	0.20	1.00	0.00	0.00	0.75000000	0.89583333	0.35416667
2	3	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.64880952	0.93103448	0.42015600
3	3	1.0	0.40	0.40	0.20	0.50	0.30	0.20	0.66071428	0.87500000	0.46428572
4	3	1.0	0.40	0.40	0.20	0.60	0.20	0.20	0.73214285	0.90384615	0.36401100
5	3	1.0	0.40	0.40	0.20	0.40	0.40	0.20	0.68452380	0.91346153	0.40201467
6	3	1.0	0.50	0.30	0.20	1.00	0.00	0.00	0.77380952	0.89583333	0.33035715
7	3	1.0	0.50	0.30	0.20	0.50	0.40	0.10	0.75000000	0.91071428	0.33928572
8	3	1.0	0.50	0.30	0.20	0.50	0.30	0.20	0.75595238	0.90384615	0.34020147
9	3	1.0	0.50	0.30	0.20	0.60	0.20	0.20	0.75595238	0.94230769	0.30173993
10	3	1.0	0.50	0.30	0.20	0.40	0.40	0.20	0.75000000	0.91071428	0.33928572
11	3	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.71666666	0.92857142	0.35476192
12	3	1.0	0.45	0.45	0.10	0.50	0.40	0.10	0.64523809	0.91666666	0.43809525
13	3	1.0	0.45	0.45	0.10	0.50	0.30	0.20	0.64523809	0.91379310	0.44096881
14	3	1.0	0.45	0.45	0.10	0.60	0.20	0.20	0.72261904	0.90740740	0.36997356
15	3	1.0	0.45	0.45	0.10	0.40	0.40	0.20	0.64523809	0.93333333	0.42142858
16	3	1.0	0.35	0.35	0.30	1.00	0.00	0.00	0.86309523	0.85000000	0.28690477
17	3	1.0	0.35	0.35	0.30	0.50	0.40	0.10	0.73809523	0.85869565	0.40320912
18	3	1.0	0.35	0.35	0.30	0.50	0.30	0.20	0.72619047	0.90384615	0.36996338
19	3	1.0	0.35	0.35	0.30	0.60	0.20	0.20	0.77976190	0.89285714	0.32738096
20	3	1.0	0.35	0.35	0.30	0.40	0.40	0.20	0.70238095	0.89333333	0.40428572
21	3	1.0	0.40	0.35	0.25	1.00	0.00	0.00	0.82142857	0.93181818	0.24675325
22	3	1.0	0.40	0.35	0.25	0.50	0.40	0.10	0.73214285	0.82954545	0.43831170
23	3	1.0	0.40	0.35	0.25	0.50	0.30	0.20	0.72619047	0.88461538	0.38919415
24	3	1.0	0.40	0.35	0.25	0.60	0.20	0.20	0.73214285	0.90384615	0.36401100
25	3	1.0	0.40	0.35	0.25	0.40	0.40	0.20	0.70238095	0.86805555	0.42956350
26	4	1.0	0.40	0.40	0.20	1.00	0.00	0.00	0.82142857	0.83333333	0.34523810
27	4	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.77023809	0.83492063	0.39484128
28	4	1.0	0.40	0.40	0.20	0.50	0.30	0.20	0.77380952	0.80555555	0.42063493
29	4	1.0	0.40	0.40	0.20	0.60	0.20	0.20	0.87500000	0.73125000	0.39375000
30	4	1.0	0.40	0.40	0.20	0.40	0.40	0.20	0.78809523	0.81060606	0.40129871
31	4	1.0	0.50	0.30	0.20	1.00	0.00	0.00	0.90476190	0.74444444	0.35079366
32	4	1.0	0.50	0.30	0.20	0.50	0.40	0.10	0.88095238	0.72857142	0.39047620
33	4	1.0	0.50	0.30	0.20	0.50	0.30	0.20	0.90714285	0.74509803	0.34775912
34	4	1.0	0.50	0.30	0.20	0.60	0.20	0.20	0.89523809	0.78645833	0.31830358
35	4	1.0	0.50	0.30	0.20	0.40	0.40	0.20	0.87142857	0.80701754	0.32155389
36	4	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.82738095	0.74166666	0.43095239
37	4	1.0	0.45	0.45	0.10	0.50	0.40	0.10	0.85357142	0.78333333	0.36309525
38	4	1.0	0.45	0.45	0.10	0.50	0.30	0.20	0.83571428	0.72444444	0.43984128
39	4	1.0	0.45	0.45	0.10	0.60	0.20	0.20	0.78809523	0.76515151	0.44675326
40	4	1.0	0.45	0.45	0.10	0.40	0.40	0.20	0.79999999	0.73333333	0.46666668
41	4	1.0	0.35	0.35	0.30	1.00	0.00	0.00	0.89523809	0.74019607	0.36456584
42	4	1.0	0.35	0.35	0.30	0.50	0.40	0.10	0.79404761	0.79924242	0.40670997
43	4	1.0	0.35	0.35	0.30	0.50	0.30	0.20	0.85714285	0.76851851	0.37433864
44	4	1.0	0.35	0.35	0.30	0.60	0.20	0.20	0.82380952	0.77310924	0.40308124
45	4	1.0	0.35	0.35	0.30	0.40	0.40	0.20	0.91666666	0.71794871	0.36538463
46	4	1.0	0.40	0.35	0.25	1.00	0.00	0.00	0.85952380	0.79166666	0.34880954
47	4	1.0	0.40	0.35	0.25	0.50	0.40	0.10	0.82380952	0.72407407	0.45211641
48	4	1.0	0.40	0.35	0.25	0.50	0.30	0.20	0.82738095	0.79166666	0.38095239
49	4	1.0	0.40	0.35	0.25	0.60	0.20	0.20	0.87142857	0.76851851	0.36005292
50	4	1.0	0.40	0.35	0.25	0.40	0.40	0.20	0.83571428	0.76960784	0.39467788
51	5	1.0	0.40	0.40	0.20	1.00	0.00	0.00	0.86309523	0.78703703	0.34986774
52	5	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.86904761	0.73412698	0.39682541
53	5	1.0	0.40	0.40	0.20	0.50	0.30	0.20	0.79761904	0.75462962	0.44775134
54	5	1.0	0.40	0.40	0.20	0.60	0.20	0.20	0.90476190	0.70555555	0.38968255
55	5	1.0	0.40	0.40	0.20	0.40	0.40	0.20	0.88095238	0.72916666	0.38988096
56	5	1.0	0.50	0.30	0.20	1.00	0.00	0.00	0.83333333	0.74074074	0.42592593
57	5	1.0	0.50	0.30	0.20	0.50	0.40	0.10	0.95238095	0.72727272	0.32034633
58	5	1.0	0.50	0.30	0.20	0.50	0.30	0.20	0.89880952	0.73333333	0.36785715
59	5	1.0	0.50	0.30	0.20	0.60	0.20	0.20	0.87142857	0.75000000	0.37857143
60	5	1.0	0.50	0.30	0.20	0.40	0.40	0.20	0.87142857	0.81481481	0.31375662
61	5	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.83333333	0.73529411	0.43137256
62	5	1.0	0.45	0.45	0.10	0.50	0.40	0.10	0.83333333	0.74479166	0.42187501
63	5	1.0	0.45	0.45	0.10	0.50	0.30	0.20	0.86904761	0.70238095	0.42857144
64	5	1.0	0.45	0.45	0.10	0.60	0.20	0.20	0.91666666	0.64444444	0.43888890
65	5	1.0	0.45	0.45	0.10	0.40	0.40	0.20	0.83333333	0.75925925	0.40740742
66	5	1.0	0.35	0.35	0.30	1.00	0.00	0.00	0.90476190	0.72916666	0.36607144
67	5	1.0	0.35	0.35	0.30	0.50	0.40	0.10	0.86904761	0.76111111	0.36984128
68	5	1.0	0.35	0.35	0.30	0.50	0.30	0.20	0.86904761	0.79629629	0.33465610
69	5	1.0	0.35	0.35	0.30	0.60	0.20	0.20	0.86904761	0.69117647	0.43977592
70	5	1.0	0.35	0.35	0.30	0.40	0.40	0.20	0.96428571	0.66666666	0.36904763
71	5	1.0	0.40	0.35	0.25	1.00	0.00	0.00	0.90476190	0.73958333	0.35565477

Continua na próxima página ...

... continuação da página anterior

n <sup>o</sup>	tp	smín	F <sub>S</sub>	F <sub>O</sub>	F <sub>N</sub>	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
72	5	1.0	0.40	0.35	0.25	0.50	0.40	0.10	0.83571428	0.71354166	0.45074406
73	5	1.0	0.40	0.35	0.25	0.50	0.30	0.20	0.86904761	0.74444444	0.38650795
74	5	1.0	0.40	0.35	0.25	0.60	0.20	0.20	0.87142857	0.77450980	0.35406163
75	5	1.0	0.40	0.35	0.25	0.40	0.40	0.20	0.83571428	0.77500000	0.38928572
76	3	0.8	0.40	0.40	0.20	1.00	0.00	0.00	0.85714285	0.88095238	0.26190482
77	3	0.8	0.40	0.40	0.20	0.50	0.40	0.10	0.76785714	0.82727272	0.40487014
78	3	0.8	0.40	0.40	0.20	0.50	0.30	0.20	0.76785714	0.86000000	0.37214286
79	3	0.8	0.40	0.40	0.20	0.60	0.20	0.20	0.76785714	0.90000000	0.33214286
80	3	0.8	0.40	0.40	0.20	0.40	0.40	0.20	0.80357142	0.85227272	0.34415586
81	3	0.8	0.50	0.30	0.20	1.00	0.00	0.00	0.77976190	0.86956521	0.35067289
82	3	0.8	0.50	0.30	0.20	0.50	0.40	0.10	0.80952380	0.84523809	0.34523811
83	3	0.8	0.50	0.30	0.20	0.50	0.30	0.20	0.76190476	0.88000000	0.35809524
84	3	0.8	0.50	0.30	0.20	0.60	0.20	0.20	0.77380952	0.91666666	0.30952382
85	3	0.8	0.50	0.30	0.20	0.40	0.40	0.20	0.76190476	0.88000000	0.35809524
86	3	0.8	0.45	0.45	0.10	1.00	0.00	0.00	0.72619047	0.90384615	0.36996338
87	3	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.72023809	0.83000000	0.44976191
88	3	0.8	0.45	0.45	0.10	0.50	0.30	0.20	0.73214285	0.81304347	0.45481368
89	3	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.71428571	0.87037037	0.41534392
90	3	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.76785714	0.86000000	0.37214286
91	3	0.8	0.35	0.35	0.30	1.00	0.00	0.00	0.86309523	0.87500000	0.26190477
92	3	0.8	0.35	0.35	0.30	0.50	0.40	0.10	0.85714285	0.75462962	0.38822753
93	3	0.8	0.35	0.35	0.30	0.50	0.30	0.20	0.85714285	0.87222222	0.27063493
94	3	0.8	0.35	0.35	0.30	0.60	0.20	0.20	0.80952380	0.81746031	0.37301589
95	3	0.8	0.35	0.35	0.30	0.40	0.40	0.20	0.82142857	0.75833333	0.42023810
96	3	0.8	0.40	0.35	0.25	1.00	0.00	0.00	0.89285714	0.88596491	0.22117795
97	3	0.8	0.40	0.35	0.25	0.50	0.40	0.10	0.80357142	0.81250000	0.38392858
98	3	0.8	0.40	0.35	0.25	0.50	0.30	0.20	0.76190476	0.90000000	0.33809524
99	3	0.8	0.40	0.35	0.25	0.60	0.20	0.20	0.73809523	0.86805555	0.39384922
100	3	0.8	0.40	0.35	0.25	0.40	0.40	0.20	0.80952380	0.83750000	0.35297620
101	4	0.8	0.40	0.40	0.20	1.00	0.00	0.00	0.89285714	0.76470588	0.34243698
102	4	0.8	0.40	0.40	0.20	0.50	0.40	0.10	0.85714285	0.76666666	0.37619049
103	4	0.8	0.40	0.40	0.20	0.50	0.30	0.20	0.89285714	0.75757575	0.34956711
104	4	0.8	0.40	0.40	0.20	0.60	0.20	0.20	0.88095238	0.73076923	0.38827839
105	4	0.8	0.40	0.40	0.20	0.40	0.40	0.20	0.91666666	0.67857142	0.40476192
106	4	0.8	0.50	0.30	0.20	1.00	0.00	0.00	0.91666666	0.64743589	0.43589745
107	4	0.8	0.50	0.30	0.20	0.50	0.40	0.10	0.91666666	0.80555555	0.27777779
108	4	0.8	0.50	0.30	0.20	0.50	0.30	0.20	0.86309523	0.74074074	0.39616403
109	4	0.8	0.50	0.30	0.20	0.60	0.20	0.20	0.87500000	0.78431372	0.34068628
110	4	0.8	0.50	0.30	0.20	0.40	0.40	0.20	0.92857142	0.74166666	0.32976192
111	4	0.8	0.45	0.45	0.10	1.00	0.00	0.00	0.86904761	0.74074074	0.39021165
112	4	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.82142857	0.72745098	0.45112045
113	4	0.8	0.45	0.45	0.10	0.50	0.30	0.20	0.82738095	0.73725490	0.43536415
114	4	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.82738095	0.79166666	0.38095239
115	4	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.87500000	0.68627450	0.43872550
116	4	0.8	0.35	0.35	0.30	1.00	0.00	0.00	0.90476190	0.70555555	0.38968255
117	4	0.8	0.35	0.35	0.30	0.50	0.40	0.10	0.90476190	0.72499999	0.37023811
118	4	0.8	0.35	0.35	0.30	0.50	0.30	0.20	0.90476190	0.73958333	0.35565477
119	4	0.8	0.35	0.35	0.30	0.60	0.20	0.20	0.91666666	0.70416666	0.37916668
120	4	0.8	0.35	0.35	0.30	0.40	0.40	0.20	0.96428571	0.66666666	0.36904763
121	4	0.8	0.40	0.35	0.25	1.00	0.00	0.00	0.90476190	0.73888888	0.35634922
122	4	0.8	0.40	0.35	0.25	0.50	0.40	0.10	0.86309523	0.63555555	0.50134922
123	4	0.8	0.40	0.35	0.25	0.50	0.30	0.20	0.91071428	0.72619047	0.36309525
124	4	0.8	0.40	0.35	0.25	0.60	0.20	0.20	0.91666666	0.72619047	0.35714287
125	4	0.8	0.40	0.35	0.25	0.40	0.40	0.20	0.87500000	0.73888888	0.38611112
126	5	0.8	0.40	0.40	0.20	1.00	0.00	0.00	0.91666666	0.69487179	0.38846155
127	5	0.8	0.40	0.40	0.20	0.50	0.40	0.10	0.86904761	0.71794871	0.41300368
128	5	0.8	0.40	0.40	0.20	0.50	0.30	0.20	0.88095238	0.63461538	0.48443224
129	5	0.8	0.40	0.40	0.20	0.60	0.20	0.20	0.91666666	0.68452380	0.39880954
130	5	0.8	0.40	0.40	0.20	0.40	0.40	0.20	0.91666666	0.68589743	0.39743591
131	5	0.8	0.50	0.30	0.20	1.00	0.00	0.00	0.84523809	0.72549019	0.42927172
132	5	0.8	0.50	0.30	0.20	0.50	0.40	0.10	0.90476190	0.68452380	0.41071430
133	5	0.8	0.50	0.30	0.20	0.50	0.30	0.20	0.91666666	0.68589743	0.39743591
134	5	0.8	0.50	0.30	0.20	0.60	0.20	0.20	0.95238095	0.72272727	0.32489178
135	5	0.8	0.50	0.30	0.20	0.40	0.40	0.20	0.88095238	0.76923076	0.34981686
136	5	0.8	0.45	0.45	0.10	1.00	0.00	0.00	0.86904761	0.71568627	0.41526612
137	5	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.83333333	0.65777777	0.50888890
138	5	0.8	0.45	0.45	0.10	0.50	0.30	0.20	0.86904761	0.70476190	0.42619049
139	5	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.84523809	0.68627450	0.46848741
140	5	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.88095238	0.66666666	0.45238096
141	5	0.8	0.35	0.35	0.30	1.00	0.00	0.00	0.90476190	0.75490196	0.34033614
142	5	0.8	0.35	0.35	0.30	0.50	0.40	0.10	0.90476190	0.75000000	0.34523810
143	5	0.8	0.35	0.35	0.30	0.50	0.30	0.20	0.90476190	0.78431372	0.31092438
144	5	0.8	0.35	0.35	0.30	0.60	0.20	0.20	0.91666666	0.66666666	0.41666668
145	5	0.8	0.35	0.35	0.30	0.40	0.40	0.20	0.91666666	0.67948717	0.40384617
146	5	0.8	0.40	0.35	0.25	1.00	0.00	0.00	0.90476190	0.72549019	0.36974791
147	5	0.8	0.40	0.35	0.25	0.50	0.40	0.10	0.87500000	0.67261904	0.45238096
148	5	0.8	0.40	0.35	0.25	0.50	0.30	0.20	0.91666666	0.70138888	0.38194446
149	5	0.8	0.40	0.35	0.25	0.60	0.20	0.20	0.91666666	0.73076923	0.35256411
150	5	0.8	0.40	0.35	0.25	0.40	0.40	0.20	0.87599999	0.75000000	0.37400001
151	3	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.63095238	0.90625000	0.46279762
152	3	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.57380952	0.93055555	0.49563493
153	3	1.2	0.40	0.40	0.20	0.50	0.30	0.20	0.57142857	0.93055555	0.49801588
154	3	1.2	0.40	0.40	0.20	0.60	0.20	0.20	0.55595238	0.94736842	0.49667920
155	3	1.2	0.40	0.40	0.20	0.40	0.40	0.20	0.57380952	0.97222222	0.45396826
156	3	1.2	0.50	0.30	0.20	1.00	0.00	0.00	0.60952380	0.92857142	0.46190478
157	3	1.2	0.50	0.30	0.20	0.50	0.40	0.10	0.65178571	0.91666666	0.43154763
158	3	1.2	0.50	0.30	0.20	0.50	0.30	0.20	0.66071428	0.90625000	0.43303572
159	3	1.2	0.50	0.30	0.20	0.60	0.20	0.20	0.65714285	0.93939393	0.40346322
160	3	1.2	0.50	0.30	0.20	0.40	0.40	0.20	0.65714285	0.92424242	0.41861473
161	3	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.47911255	0.94897959	0.57190786
162	3	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.49101731	0.95833333	0.55064936

Continua na próxima página ...

... continuação da página anterior

n°	tp	smín	FS	FO	FN	FN <sub>M</sub>	FN <sub>C</sub>	FPR	Dispersão	Diversificação	Diferença
163	3	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.49101731	0.94680851	0.56217418
164	3	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.47911255	0.93877551	0.58211194
165	3	1.2	0.45	0.45	0.10	0.40	0.40	0.20	0.49345238	0.95555555	0.55099207
166	3	1.2	0.35	0.35	0.30	1.00	0.00	0.00	0.62329931	0.94285714	0.43384355
167	3	1.2	0.35	0.35	0.30	0.50	0.40	0.10	0.64285714	0.91935483	0.43778803
168	3	1.2	0.35	0.35	0.30	0.50	0.30	0.20	0.63095238	0.93750000	0.43154762
169	3	1.2	0.35	0.35	0.30	0.60	0.20	0.20	0.66666666	0.91935483	0.41397851
170	3	1.2	0.35	0.35	0.30	0.40	0.40	0.20	0.67499999	0.93548387	0.38951614
171	3	1.2	0.40	0.35	0.25	1.00	0.00	0.00	0.63095238	0.93750000	0.43154762
172	3	1.2	0.40	0.35	0.25	0.50	0.40	0.10	0.63928571	0.89655172	0.46416257
173	3	1.2	0.40	0.35	0.25	0.50	0.30	0.20	0.62380952	0.93939393	0.43679655
174	3	1.2	0.40	0.35	0.25	0.60	0.20	0.20	0.60654761	0.93589743	0.45755496
175	3	1.2	0.40	0.35	0.25	0.40	0.40	0.20	0.64523809	0.91379310	0.44096881
176	4	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.69710884	0.89583333	0.40705783
177	4	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.75238095	0.85555555	0.39206350
178	4	1.2	0.40	0.40	0.20	0.50	0.30	0.20	0.71666666	0.84259259	0.44074075
179	4	1.2	0.40	0.40	0.20	0.60	0.20	0.20	0.71428571	0.60978835	0.67592594
180	4	1.2	0.40	0.40	0.20	0.40	0.40	0.20	0.80952380	0.78070175	0.40977445
181	4	1.2	0.50	0.30	0.20	1.00	0.00	0.00	0.79047619	0.80902777	0.40049604
182	4	1.2	0.50	0.30	0.20	0.50	0.40	0.10	0.76428571	0.85333333	0.38238096
183	4	1.2	0.50	0.30	0.20	0.50	0.30	0.20	0.84761904	0.80303030	0.34935066
184	4	1.2	0.50	0.30	0.20	0.60	0.20	0.20	0.74642857	0.86419753	0.38937390
185	4	1.2	0.50	0.30	0.20	0.40	0.40	0.20	0.81190476	0.86805555	0.32003969
186	4	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.73452380	0.89784946	0.36762674
187	4	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.74472789	0.86666666	0.38860545
188	4	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.69880952	0.82795698	0.47323350
189	4	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.68690476	0.84895833	0.46413691
190	4	1.2	0.45	0.45	0.10	0.40	0.40	0.20	0.80595238	0.75757575	0.43647187
191	4	1.2	0.35	0.35	0.30	1.00	0.00	0.00	0.78809523	0.83012820	0.38177657
192	4	1.2	0.35	0.35	0.30	0.50	0.40	0.10	0.79404761	0.78985507	0.41609732
193	4	1.2	0.35	0.35	0.30	0.50	0.30	0.20	0.85714285	0.75208333	0.39077382
194	4	1.2	0.35	0.35	0.30	0.60	0.20	0.20	0.70476190	0.83908045	0.45615765
195	4	1.2	0.35	0.35	0.30	0.40	0.40	0.20	0.88095238	0.80000000	0.31904762
196	4	1.2	0.40	0.35	0.25	1.00	0.00	0.00	0.76428571	0.83974358	0.39597071
197	4	1.2	0.40	0.35	0.25	0.50	0.40	0.10	0.82380952	0.67843137	0.49775911
198	4	1.2	0.40	0.35	0.25	0.50	0.30	0.20	0.82380952	0.76754385	0.40864663
199	4	1.2	0.40	0.35	0.25	0.60	0.20	0.20	0.77619047	0.82638888	0.39742065
200	4	1.2	0.40	0.35	0.25	0.40	0.40	0.20	0.87142857	0.75000000	0.37857143
201	5	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.75068027	0.88505747	0.36426226
202	5	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.74642857	0.86133333	0.39223810
203	5	1.2	0.40	0.40	0.20	0.50	0.30	0.20	0.71666666	0.87931034	0.40402300
204	5	1.2	0.40	0.40	0.20	0.60	0.20	0.20	0.73452380	0.83908045	0.42639575
205	5	1.2	0.40	0.40	0.20	0.40	0.40	0.20	0.79761904	0.77192982	0.43045114
206	5	1.2	0.50	0.30	0.20	1.00	0.00	0.00	0.80952380	0.76984126	0.42063494
207	5	1.2	0.50	0.30	0.20	0.50	0.40	0.10	0.76428571	0.84615384	0.38956045
208	5	1.2	0.50	0.30	0.20	0.50	0.30	0.20	0.88928571	0.74561403	0.36510026
209	5	1.2	0.50	0.30	0.20	0.60	0.20	0.20	0.83571428	0.86111111	0.30317461
210	5	1.2	0.50	0.30	0.20	0.40	0.40	0.20	0.81190476	0.86805555	0.32003969
211	5	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.74642857	0.82098765	0.43258378
212	5	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.73877551	0.88172043	0.37950406
213	5	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.72261904	0.80864197	0.46873899
214	5	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.71071428	0.85632183	0.43296389
215	5	1.2	0.45	0.45	0.10	0.40	0.40	0.20	0.77023809	0.80555555	0.42420636
216	5	1.2	0.35	0.35	0.30	1.00	0.00	0.00	0.78809523	0.83012820	0.38177657
217	5	1.2	0.35	0.35	0.30	0.50	0.40	0.10	0.81190476	0.78787878	0.40021646
218	5	1.2	0.35	0.35	0.30	0.50	0.30	0.20	0.80952380	0.81060606	0.37987014
219	5	1.2	0.35	0.35	0.30	0.60	0.20	0.20	0.75238095	0.81730769	0.43031136
220	5	1.2	0.35	0.35	0.30	0.40	0.40	0.20	0.88095238	0.80000000	0.31904762
221	5	1.2	0.40	0.35	0.25	1.00	0.00	0.00	0.75238095	0.89080459	0.35681446
222	5	1.2	0.40	0.35	0.25	0.50	0.40	0.10	0.82380952	0.67843137	0.49775911
223	5	1.2	0.40	0.35	0.25	0.50	0.30	0.20	0.76428571	0.82666666	0.40904763
224	5	1.2	0.40	0.35	0.25	0.60	0.20	0.20	0.77619047	0.82638888	0.39742065
225	5	1.2	0.40	0.35	0.25	0.40	0.40	0.20	0.87142857	0.72395833	0.40461310

Tabela B.53: Calibração da Instância 18: Tomcat

n°	tp	smín	FS	FO	FN	FN <sub>M</sub>	FN <sub>C</sub>	FPR	Dispersão	Diversificação	Diferença
1	3	1.0	0.40	0.40	0.20	1.00	0.00	0.00	0.87592592	0.82040816	0.30366591
2	3	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.86461640	0.81066666	0.32471693
3	3	1.0	0.40	0.40	0.20	0.50	0.30	0.20	0.87669753	0.83717948	0.28612298
4	3	1.0	0.40	0.40	0.20	0.60	0.20	0.20	0.89328703	0.81629629	0.29041666
5	3	1.0	0.40	0.40	0.20	0.40	0.40	0.20	0.87757936	0.81066666	0.31175396
6	3	1.0	0.50	0.30	0.20	1.00	0.00	0.00	0.88865740	0.87333333	0.23800925
7	3	1.0	0.50	0.30	0.20	0.50	0.40	0.10	0.87824074	0.84294871	0.27881054
8	3	1.0	0.50	0.30	0.20	0.50	0.30	0.20	0.87554012	0.86781609	0.25664378
9	3	1.0	0.50	0.30	0.20	0.60	0.20	0.20	0.90933661	0.86728395	0.22337962
10	3	1.0	0.50	0.30	0.20	0.40	0.40	0.20	0.87824074	0.83333333	0.28842592
11	3	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.85234788	0.82816091	0.31949119
12	3	1.0	0.45	0.45	0.10	0.50	0.40	0.10	0.83827160	0.83988095	0.32184744
13	3	1.0	0.45	0.45	0.10	0.50	0.30	0.20	0.83827160	0.84210526	0.31962313
14	3	1.0	0.45	0.45	0.10	0.60	0.20	0.20	0.85352564	0.86071428	0.28576007
15	3	1.0	0.45	0.45	0.10	0.40	0.40	0.20	0.81648860	0.88557213	0.29793925
16	3	1.0	0.35	0.35	0.30	1.00	0.00	0.00	0.90026455	0.86728395	0.23245149
17	3	1.0	0.35	0.35	0.30	0.50	0.40	0.10	0.89351851	0.83647798	0.27000349
18	3	1.0	0.35	0.35	0.30	0.50	0.30	0.20	0.89097222	0.78061224	0.32841553
19	3	1.0	0.35	0.35	0.30	0.60	0.20	0.20	0.88544973	0.82666666	0.28788359
20	3	1.0	0.35	0.35	0.30	0.40	0.40	0.20	0.91203703	0.80277777	0.28518518
21	3	1.0	0.40	0.35	0.25	1.00	0.00	0.00	0.88101851	0.88787878	0.23110269
22	3	1.0	0.40	0.35	0.25	0.50	0.40	0.10	0.89146825	0.82653061	0.28200113
23	3	1.0	0.40	0.35	0.25	0.50	0.30	0.20	0.87850529	0.84294871	0.27854599
24	3	1.0	0.40	0.35	0.25	0.60	0.20	0.20	0.91759259	0.84751773	0.23488967

Continua na próxima página ...



... continuação da página anterior

n°	tp	smín	FS	FO	FN	FN <sub>M</sub>	FN <sub>C</sub>	FP <sub>R</sub>	Dispersão	Diversificação	Diferença
25	3	1.0	0.40	0.35	0.25	0.40	0.40	0.20	0.88287037	0.78260869	0.33452093
26	4	1.0	0.40	0.40	0.20	1.00	0.00	0.00	0.91365741	0.78949275	0.29684984
27	4	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.92129630	0.70555556	0.37314815
28	4	1.0	0.40	0.40	0.20	0.50	0.30	0.20	0.92361111	0.64830918	0.42807971
29	4	1.0	0.40	0.40	0.20	0.60	0.20	0.20	0.92453704	0.68173077	0.39373219
30	4	1.0	0.40	0.40	0.20	0.40	0.40	0.20	0.90972222	0.66285714	0.42742063
31	4	1.0	0.50	0.30	0.20	1.00	0.00	0.00	0.91643519	0.76219512	0.32136969
32	4	1.0	0.50	0.30	0.20	0.50	0.40	0.10	0.89054233	0.78888889	0.32056878
33	4	1.0	0.50	0.30	0.20	0.50	0.30	0.20	0.91550926	0.73859649	0.34589425
34	4	1.0	0.50	0.30	0.20	0.60	0.20	0.20	0.93915344	0.70613027	0.35471629
35	4	1.0	0.50	0.30	0.20	0.40	0.40	0.20	0.91600529	0.74978632	0.33420838
36	4	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.93865741	0.73166667	0.32967593
37	4	1.0	0.45	0.45	0.10	0.50	0.40	0.10	0.88425926	0.76222651	0.35351423
38	4	1.0	0.45	0.45	0.10	0.50	0.30	0.20	0.87268519	0.75813008	0.36918473
39	4	1.0	0.45	0.45	0.10	0.60	0.20	0.20	0.88888889	0.75611326	0.35499785
40	4	1.0	0.45	0.45	0.10	0.40	0.40	0.20	0.88888889	0.76250000	0.34861111
41	4	1.0	0.35	0.35	0.30	1.00	0.00	0.00	0.91446208	0.82826577	0.25727215
42	4	1.0	0.35	0.35	0.30	0.50	0.40	0.10	0.93683862	0.69244792	0.37071346
43	4	1.0	0.35	0.35	0.30	0.50	0.30	0.20	0.91388889	0.68873874	0.39737237
44	4	1.0	0.35	0.35	0.30	0.60	0.20	0.20	0.92063492	0.67903226	0.40033282
45	4	1.0	0.35	0.35	0.30	0.40	0.40	0.20	0.93981481	0.63277778	0.42740741
46	4	1.0	0.40	0.35	0.25	1.00	0.00	0.00	0.91693122	0.75252525	0.33054353
47	4	1.0	0.40	0.35	0.25	0.50	0.40	0.10	0.93981481	0.62623457	0.43395062
48	4	1.0	0.40	0.35	0.25	0.50	0.30	0.20	0.90000000	0.71359649	0.38640351
49	4	1.0	0.40	0.35	0.25	0.60	0.20	0.20	0.92129630	0.76018519	0.31851852
50	4	1.0	0.40	0.35	0.25	0.40	0.40	0.20	0.93981481	0.62142857	0.43875661
51	5	1.0	0.40	0.40	0.20	1.00	0.00	0.00	0.93472222	0.72321937	0.34205840
52	5	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.91759259	0.62611111	0.45629630
53	5	1.0	0.40	0.40	0.20	0.50	0.30	0.20	0.95925926	0.61365741	0.42708333
54	5	1.0	0.40	0.40	0.20	0.60	0.20	0.20	0.94675926	0.67466667	0.37857407
55	5	1.0	0.40	0.40	0.20	0.40	0.40	0.20	0.91759259	0.64092593	0.44148148
56	5	1.0	0.50	0.30	0.20	1.00	0.00	0.00	0.93240741	0.71936937	0.34822322
57	5	1.0	0.50	0.30	0.20	0.50	0.40	0.10	0.90740741	0.68783784	0.40475475
58	5	1.0	0.50	0.30	0.20	0.50	0.30	0.20	0.92129630	0.71666667	0.36203704
59	5	1.0	0.50	0.30	0.20	0.60	0.20	0.20	0.94444444	0.64166667	0.41388889
60	5	1.0	0.50	0.30	0.20	0.40	0.40	0.20	0.90740741	0.70042735	0.39216524
61	5	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.92083333	0.72017544	0.35899123
62	5	1.0	0.45	0.45	0.10	0.50	0.40	0.10	0.90509259	0.63850733	0.45640008
63	5	1.0	0.45	0.45	0.10	0.50	0.30	0.20	0.90462963	0.71936275	0.37600763
64	5	1.0	0.45	0.45	0.10	0.60	0.20	0.20	0.91435185	0.70703125	0.37861690
65	5	1.0	0.45	0.45	0.10	0.40	0.40	0.20	0.89675926	0.67117117	0.43206957
66	5	1.0	0.35	0.35	0.30	1.00	0.00	0.00	0.93263889	0.73245614	0.33490497
67	5	1.0	0.35	0.35	0.30	0.50	0.40	0.10	0.92685185	0.62787356	0.44527458
68	5	1.0	0.35	0.35	0.30	0.50	0.30	0.20	0.92824074	0.62049689	0.45126236
69	5	1.0	0.35	0.35	0.30	0.60	0.20	0.20	0.92314815	0.66075269	0.41609916
70	5	1.0	0.35	0.35	0.30	0.40	0.40	0.20	0.94074074	0.62011494	0.43914432
71	5	1.0	0.40	0.35	0.25	1.00	0.00	0.00	0.93472222	0.72247475	0.34280303
72	5	1.0	0.40	0.35	0.25	0.50	0.40	0.10	0.91759259	0.68198198	0.40042543
73	5	1.0	0.40	0.35	0.25	0.50	0.30	0.20	0.91759259	0.65106838	0.43133903
74	5	1.0	0.40	0.35	0.25	0.60	0.20	0.20	0.92453704	0.71209677	0.36336619
75	5	1.0	0.40	0.35	0.25	0.40	0.40	0.20	0.94444444	0.62253968	0.43301587
76	3	0.8	0.40	0.40	0.20	1.00	0.00	0.00	0.91527778	0.80217687	0.28254535
77	3	0.8	0.40	0.40	0.20	0.50	0.40	0.10	0.95396825	0.71590909	0.33012266
78	3	0.8	0.40	0.40	0.20	0.50	0.30	0.20	0.93240741	0.78108466	0.28650794
79	3	0.8	0.40	0.40	0.20	0.60	0.20	0.20	0.91759259	0.78157895	0.30082846
80	3	0.8	0.40	0.40	0.20	0.40	0.40	0.20	0.93313492	0.74122807	0.32563701
81	3	0.8	0.50	0.30	0.20	1.00	0.00	0.00	0.90231481	0.86124031	0.23644488
82	3	0.8	0.50	0.30	0.20	0.50	0.40	0.10	0.91990741	0.81851852	0.26157407
83	3	0.8	0.50	0.30	0.20	0.50	0.30	0.20	0.91759259	0.81185185	0.27055556
84	3	0.8	0.50	0.30	0.20	0.60	0.20	0.20	0.93287037	0.83333333	0.23379630
85	3	0.8	0.50	0.30	0.20	0.40	0.40	0.20	0.91990741	0.80740741	0.27268519
86	3	0.8	0.45	0.45	0.10	1.00	0.00	0.00	0.91643519	0.79347826	0.29008655
87	3	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.89290123	0.80252976	0.30456900
88	3	0.8	0.45	0.45	0.10	0.50	0.30	0.20	0.88842593	0.77857143	0.33300265
89	3	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.88611111	0.81028694	0.30360195
90	3	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.88452381	0.81371882	0.30175737
91	3	0.8	0.35	0.35	0.30	1.00	0.00	0.00	0.90277778	0.80306268	0.29415954
92	3	0.8	0.35	0.35	0.30	0.50	0.40	0.10	0.91296296	0.77182540	0.31521164
93	3	0.8	0.35	0.35	0.30	0.50	0.30	0.20	0.90138889	0.77243590	0.32617521
94	3	0.8	0.35	0.35	0.30	0.60	0.20	0.20	0.93703704	0.76031870	0.30264426
95	3	0.8	0.35	0.35	0.30	0.40	0.40	0.20	0.91990741	0.76064214	0.31945046
96	3	0.8	0.40	0.35	0.25	1.00	0.00	0.00	0.90509259	0.80833333	0.28657407
97	3	0.8	0.40	0.35	0.25	0.50	0.40	0.10	0.88518519	0.81111111	0.30370370
98	3	0.8	0.40	0.35	0.25	0.50	0.30	0.20	0.93703704	0.76190476	0.30105820
99	3	0.8	0.40	0.35	0.25	0.60	0.20	0.20	0.93472222	0.79426129	0.27101648
100	3	0.8	0.40	0.35	0.25	0.40	0.40	0.20	0.94212963	0.70303030	0.35484007
101	4	0.8	0.40	0.40	0.20	1.00	0.00	0.00	0.91851852	0.80398551	0.27749597
102	4	0.8	0.40	0.40	0.20	0.50	0.40	0.10	0.95396825	0.69022989	0.35580186
103	4	0.8	0.40	0.40	0.20	0.50	0.30	0.20	0.95833333	0.73469388	0.30697279
104	4	0.8	0.40	0.40	0.20	0.60	0.20	0.20	0.94537037	0.63846154	0.41616809
105	4	0.8	0.40	0.40	0.20	0.40	0.40	0.20	0.95833333	0.59837963	0.44328704
106	4	0.8	0.50	0.30	0.20	1.00	0.00	0.00	0.91990741	0.74973118	0.33036141
107	4	0.8	0.50	0.30	0.20	0.50	0.40	0.10	0.93240741	0.73991228	0.32768031
108	4	0.8	0.50	0.30	0.20	0.50	0.30	0.20	0.94212963	0.67150538	0.38636499
109	4	0.8	0.50	0.30	0.20	0.60	0.20	0.20	0.94537037	0.68555556	0.36907407
110	4	0.8	0.50	0.30	0.20	0.40	0.40	0.20	0.91851852	0.71504630	0.36643519
111	4	0.8	0.45	0.45	0.10	1.00	0.00	0.00	0.93313492	0.73557423	0.33129085
112	4	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.93055556	0.67857143	0.39087302
113	4	0.8	0.45	0.45	0.10	0.50	0.30	0.20	0.91064815	0.70459770	0.38475415
114	4	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.90046296	0.70534979	0.39418724
115	4	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.90277778	0.73024691	0.36697531

Continua na próxima página ...

... continuação da página anterior												
n°	tp	smín	FS	FO	FN	FNM	ENC	FP	Dispersão	Diversificação	Diferença	
116	4	0.8	0.35	0.35	0.30	1.00	0.00	0.00	0.92453704	0.80406746	0.27139550	
117	4	0.8	0.35	0.35	0.30	0.50	0.40	0.10	0.92222222	0.68548535	0.39229243	
118	4	0.8	0.35	0.35	0.30	0.50	0.30	0.20	0.97916667	0.52282913	0.49800420	
119	4	0.8	0.35	0.35	0.30	0.60	0.20	0.20	0.94675926	0.60802469	0.44521605	
120	4	0.8	0.35	0.35	0.30	0.40	0.40	0.20	0.94907407	0.59911111	0.45181481	
121	4	0.8	0.40	0.35	0.25	1.00	0.00	0.00	0.93842593	0.72568627	0.33588780	
122	4	0.8	0.40	0.35	0.25	0.50	0.40	0.10	0.94907407	0.56987179	0.48105413	
123	4	0.8	0.40	0.35	0.25	0.50	0.30	0.20	0.95601852	0.65427350	0.38970798	
124	4	0.8	0.40	0.35	0.25	0.60	0.20	0.20	0.94444444	0.67916667	0.37638889	
125	4	0.8	0.40	0.35	0.25	0.40	0.40	0.20	0.96759259	0.55353535	0.47887205	
126	5	0.8	0.40	0.40	0.20	1.00	0.00	0.00	0.93842593	0.65714031	0.40443376	
127	5	0.8	0.40	0.40	0.20	0.50	0.40	0.10	0.96759259	0.52250000	0.50990741	
128	5	0.8	0.40	0.40	0.20	0.50	0.30	0.20	0.95925926	0.59299517	0.44774557	
129	5	0.8	0.40	0.40	0.20	0.60	0.20	0.20	0.94675926	0.63000000	0.42324074	
130	5	0.8	0.40	0.40	0.20	0.40	0.40	0.20	0.95925926	0.59485597	0.44588477	
131	5	0.8	0.50	0.30	0.20	1.00	0.00	0.00	0.93472222	0.73773946	0.32753831	
132	5	0.8	0.50	0.30	0.20	0.50	0.40	0.10	0.91898148	0.68648649	0.39453203	
133	5	0.8	0.50	0.30	0.20	0.50	0.30	0.20	0.92592593	0.64947090	0.42460317	
134	5	0.8	0.50	0.30	0.20	0.60	0.20	0.20	0.94907407	0.60978261	0.44114332	
135	5	0.8	0.50	0.30	0.20	0.40	0.40	0.20	0.92129630	0.67833333	0.40037037	
136	5	0.8	0.45	0.45	0.10	1.00	0.00	0.00	0.93333333	0.71466667	0.35200000	
137	5	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.91898148	0.60645604	0.47456247	
138	5	0.8	0.45	0.45	0.10	0.50	0.30	0.20	0.93518519	0.58712121	0.47769360	
139	5	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.93750000	0.61574074	0.44675926	
140	5	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.90972222	0.68682796	0.40344982	
141	5	0.8	0.35	0.35	0.30	1.00	0.00	0.00	0.93842593	0.72314815	0.33842593	
142	5	0.8	0.35	0.35	0.30	0.50	0.40	0.10	0.94907407	0.53164251	0.51928341	
143	5	0.8	0.35	0.35	0.30	0.50	0.30	0.20	0.96527778	0.51991793	0.51480429	
144	5	0.8	0.35	0.35	0.30	0.60	0.20	0.20	0.95925926	0.57266667	0.46807407	
145	5	0.8	0.35	0.35	0.30	0.40	0.40	0.20	0.91990741	0.64656863	0.43352397	
146	5	0.8	0.40	0.35	0.25	1.00	0.00	0.00	0.93750000	0.68472222	0.37777778	
147	5	0.8	0.40	0.35	0.25	0.50	0.40	0.10	0.92361111	0.62194444	0.45444444	
148	5	0.8	0.40	0.35	0.25	0.50	0.30	0.20	0.95925926	0.62330827	0.41743247	
149	5	0.8	0.40	0.35	0.25	0.60	0.20	0.20	0.92824074	0.71923077	0.35252849	
150	5	0.8	0.40	0.35	0.25	0.40	0.40	0.20	0.94212963	0.60952381	0.44834656	
151	3	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.82269880	0.90444444	0.27285675	
152	3	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.84043210	0.83602151	0.32354640	
153	3	1.2	0.40	0.40	0.20	0.50	0.30	0.20	0.84908610	0.87037037	0.28054353	
154	3	1.2	0.40	0.40	0.20	0.60	0.20	0.20	0.84953704	0.86158192	0.28888104	
155	3	1.2	0.40	0.40	0.20	0.40	0.40	0.20	0.85470679	0.82514620	0.32014701	
156	3	1.2	0.50	0.30	0.20	1.00	0.00	0.00	0.84421296	0.90049751	0.25528952	
157	3	1.2	0.50	0.30	0.20	0.50	0.40	0.10	0.84517196	0.86309524	0.29173280	
158	3	1.2	0.50	0.30	0.20	0.50	0.30	0.20	0.85570988	0.85792350	0.28636663	
159	3	1.2	0.50	0.30	0.20	0.60	0.20	0.20	0.87924383	0.87202381	0.24873236	
160	3	1.2	0.50	0.30	0.20	0.40	0.40	0.20	0.84517196	0.85416667	0.30066138	
161	3	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.84104938	0.85792350	0.30102712	
162	3	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.81795635	0.84825871	0.33378494	
163	3	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.81795635	0.87254902	0.30949463	
164	3	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.83292379	0.88512821	0.28194801	
165	3	1.2	0.45	0.45	0.10	0.40	0.40	0.20	0.81849747	0.84895833	0.33254419	
166	3	1.2	0.35	0.35	0.30	1.00	0.00	0.00	0.84561388	0.87830688	0.27607924	
167	3	1.2	0.35	0.35	0.30	0.50	0.40	0.10	0.84976852	0.86338798	0.28684350	
168	3	1.2	0.35	0.35	0.30	0.50	0.30	0.20	0.85023148	0.82424242	0.32552609	
169	3	1.2	0.35	0.35	0.30	0.60	0.20	0.20	0.84864418	0.83918129	0.31217453	
170	3	1.2	0.35	0.35	0.30	0.40	0.40	0.20	0.84884259	0.87037037	0.28078704	
171	3	1.2	0.40	0.35	0.25	1.00	0.00	0.00	0.84745370	0.89722222	0.25532407	
172	3	1.2	0.40	0.35	0.25	0.50	0.40	0.10	0.85466270	0.86549708	0.27984023	
173	3	1.2	0.40	0.35	0.25	0.50	0.30	0.20	0.86165123	0.87643678	0.26191198	
174	3	1.2	0.40	0.35	0.25	0.60	0.20	0.20	0.85702160	0.87853107	0.26444732	
175	3	1.2	0.40	0.35	0.25	0.40	0.40	0.20	0.85162037	0.83040936	0.31797027	
176	4	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.86700838	0.79938272	0.33360891	
177	4	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.90740741	0.69619048	0.39640212	
178	4	1.2	0.40	0.40	0.20	0.50	0.30	0.20	0.93981481	0.64583333	0.41435185	
179	4	1.2	0.40	0.40	0.20	0.60	0.20	0.20	0.92453704	0.68697917	0.38848380	
180	4	1.2	0.40	0.40	0.20	0.40	0.40	0.20	0.90601852	0.71333333	0.38064815	
181	4	1.2	0.50	0.30	0.20	1.00	0.00	0.00	0.89938272	0.79421769	0.30639960	
182	4	1.2	0.50	0.30	0.20	0.50	0.40	0.10	0.87136243	0.81111111	0.31752646	
183	4	1.2	0.50	0.30	0.20	0.50	0.30	0.20	0.91550926	0.73991228	0.34457846	
184	4	1.2	0.50	0.30	0.20	0.60	0.20	0.20	0.92526455	0.70931373	0.36542172	
185	4	1.2	0.50	0.30	0.20	0.40	0.40	0.20	0.87599206	0.79425926	0.32974868	
186	4	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.87202381	0.75925926	0.36871693	
187	4	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.88271605	0.78294574	0.33433821	
188	4	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.87500000	0.75000000	0.37500000	
189	4	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.86882716	0.79444444	0.33672840	
190	4	1.2	0.45	0.45	0.10	0.40	0.40	0.20	0.86882716	0.75992063	0.37125220	
191	4	1.2	0.35	0.35	0.30	1.00	0.00	0.00	0.89885462	0.80035461	0.30079077	
192	4	1.2	0.35	0.35	0.30	0.50	0.40	0.10	0.91319444	0.71965812	0.36714744	
193	4	1.2	0.35	0.35	0.30	0.50	0.30	0.20	0.92083333	0.70000000	0.37916667	
194	4	1.2	0.35	0.35	0.30	0.60	0.20	0.20	0.89891975	0.76063830	0.34041195	
195	4	1.2	0.35	0.35	0.30	0.40	0.40	0.20	0.91319444	0.73589744	0.35090812	
196	4	1.2	0.40	0.35	0.25	1.00	0.00	0.00	0.90069444	0.79583333	0.30347222	
197	4	1.2	0.40	0.35	0.25	0.50	0.40	0.10	0.89328704	0.73914729	0.36756568	
198	4	1.2	0.40	0.35	0.25	0.50	0.30	0.20	0.89652778	0.72613636	0.37733586	
199	4	1.2	0.40	0.35	0.25	0.60	0.20	0.20	0.90023148	0.76625000	0.33351852	
200	4	1.2	0.40	0.35	0.25	0.40	0.40	0.20	0.88750000	0.69658120	0.41591880	
201	5	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.92873677	0.76851852	0.30274471	
202	5	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.93981481	0.61506410	0.44512108	
203	5	1.2	0.40	0.40	0.20	0.50	0.30	0.20	0.91898148	0.73387097	0.34714755	
204	5	1.2	0.40	0.40	0.20	0.60	0.20	0.20	0.92526455	0.69925926	0.37547619	
205	5	1.2	0.40	0.40	0.20	0.40	0.40	0.20	0.93148148	0.65059524	0.41792328	
206	5	1.2	0.50	0.30	0.20	1.00	0.00	0.00	0.91484788	0.75852713	0.32662498	

Continua na próxima página ...

... continuação da página anterior

$n^\circ$	$tp$	$s_{min}$	$F_S$	$F_O$	$F_N$	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
207	5	1.2	0.50	0.30	0.20	0.50	0.40	0.10	0.88591270	0.74883721	0.36525009
208	5	1.2	0.50	0.30	0.20	0.50	0.30	0.20	0.94444444	0.67500000	0.38055556
209	5	1.2	0.50	0.30	0.20	0.60	0.20	0.20	0.92526455	0.69714286	0.37759259
210	5	1.2	0.50	0.30	0.20	0.40	0.40	0.20	0.88591270	0.73992248	0.37416482
211	5	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.91577381	0.79166667	0.29255952
212	5	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.89429012	0.74166667	0.36404321
213	5	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.88425926	0.70857008	0.40717066
214	5	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.90159933	0.75277778	0.34562290
215	5	1.2	0.45	0.45	0.10	0.40	0.40	0.20	0.88055556	0.70476190	0.41468254
216	5	1.2	0.35	0.35	0.30	1.00	0.00	0.00	0.91626984	0.78143275	0.30229741
217	5	1.2	0.35	0.35	0.30	0.50	0.40	0.10	0.88425926	0.68000000	0.43574074
218	5	1.2	0.35	0.35	0.30	0.50	0.30	0.20	0.91435185	0.65161290	0.43403524
219	5	1.2	0.35	0.35	0.30	0.60	0.20	0.20	0.92106481	0.66851852	0.41041667
220	5	1.2	0.35	0.35	0.30	0.40	0.40	0.20	0.91666667	0.70095238	0.38238095
221	5	1.2	0.40	0.35	0.25	1.00	0.00	0.00	0.91418651	0.77296296	0.31285053
222	5	1.2	0.40	0.35	0.25	0.50	0.40	0.10	0.89814815	0.70683761	0.39501425
223	5	1.2	0.40	0.35	0.25	0.50	0.30	0.20	0.90023148	0.72439024	0.37537827
224	5	1.2	0.40	0.35	0.25	0.60	0.20	0.20	0.90211640	0.75940171	0.33848189
225	5	1.2	0.40	0.35	0.25	0.40	0.40	0.20	0.88888889	0.67264957	0.43846154

Tabela B.54: Calibração da Instância 18: HSQLDB

$n^\circ$	$tp$	$s_{min}$	$F_S$	$F_O$	$F_N$	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
1	3	1.0	0.40	0.40	0.20	1.00	0.00	0.00	0.96428571	0.76666667	0.26904762
2	3	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.89285714	0.58333333	0.52380952
3	3	1.0	0.40	0.40	0.20	0.50	0.30	0.20	0.92857143	0.71875000	0.35267857
4	3	1.0	0.40	0.40	0.20	0.60	0.20	0.20	1.00000000	0.74545455	0.25454545
5	3	1.0	0.40	0.40	0.20	0.40	0.40	0.20	0.89285714	0.64705882	0.46008403
6	3	1.0	0.50	0.30	0.20	1.00	0.00	0.00	0.96428571	0.76428571	0.27142857
7	3	1.0	0.50	0.30	0.20	0.50	0.40	0.10	0.89285714	0.73958333	0.36755952
8	3	1.0	0.50	0.30	0.20	0.50	0.30	0.20	0.92857143	0.78125000	0.29017857
9	3	1.0	0.50	0.30	0.20	0.60	0.20	0.20	0.96428571	0.80000000	0.23571429
10	3	1.0	0.50	0.30	0.20	0.40	0.40	0.20	0.92857143	0.66666667	0.40476190
11	3	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.92857143	0.68750000	0.38392857
12	3	1.0	0.45	0.45	0.10	0.50	0.40	0.10	0.89285714	0.64705882	0.46008403
13	3	1.0	0.45	0.45	0.10	0.50	0.30	0.20	0.89285714	0.64705882	0.46008403
14	3	1.0	0.45	0.45	0.10	0.60	0.20	0.20	0.89285714	0.64705882	0.46008403
15	3	1.0	0.45	0.45	0.10	0.40	0.40	0.20	0.89285714	0.64705882	0.46008403
16	3	1.0	0.35	0.35	0.30	1.00	0.00	0.00	0.96428571	0.71428571	0.32142857
17	3	1.0	0.35	0.35	0.30	0.50	0.40	0.10	0.92857143	0.73214286	0.33928571
18	3	1.0	0.35	0.35	0.30	0.50	0.30	0.20	0.96428571	0.73809524	0.29761905
19	3	1.0	0.35	0.35	0.30	0.60	0.20	0.20	0.96428571	0.76666667	0.26904762
20	3	1.0	0.35	0.35	0.30	0.40	0.40	0.20	0.92857143	0.66025641	0.41117216
21	3	1.0	0.40	0.35	0.25	1.00	0.00	0.00	0.96428571	0.83333333	0.20238095
22	3	1.0	0.40	0.35	0.25	0.50	0.40	0.10	0.92857143	0.72222222	0.34920635
23	3	1.0	0.40	0.35	0.25	0.50	0.30	0.20	0.96428571	0.71153846	0.32417582
24	3	1.0	0.40	0.35	0.25	0.60	0.20	0.20	0.96428571	0.68055556	0.35515873
25	3	1.0	0.40	0.35	0.25	0.40	0.40	0.20	0.92857143	0.68750000	0.38392857
26	4	1.0	0.40	0.40	0.20	1.00	0.00	0.00	0.96428571	0.62820513	0.40750916
27	4	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.92857143	0.44242424	0.62900433
28	4	1.0	0.40	0.40	0.20	0.50	0.30	0.20	1.00000000	0.46527778	0.53472222
29	4	1.0	0.40	0.40	0.20	0.60	0.20	0.20	1.00000000	0.47916667	0.52083333
30	4	1.0	0.40	0.40	0.20	0.40	0.40	0.20	1.00000000	0.40136054	0.59863946
31	4	1.0	0.50	0.30	0.20	1.00	0.00	0.00	1.00000000	0.49702381	0.50297619
32	4	1.0	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.49074074	0.50925926
33	4	1.0	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.48148148	0.51851852
34	4	1.0	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.47916667	0.52083333
35	4	1.0	0.50	0.30	0.20	0.40	0.40	0.20	1.00000000	0.42261905	0.57738095
36	4	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.92857143	0.47883598	0.59259259
37	4	1.0	0.45	0.45	0.10	0.50	0.40	0.10	1.00000000	0.41666667	0.58333333
38	4	1.0	0.45	0.45	0.10	0.50	0.30	0.20	1.00000000	0.41666667	0.58333333
39	4	1.0	0.45	0.45	0.10	0.60	0.20	0.20	1.00000000	0.38194444	0.61805556
40	4	1.0	0.45	0.45	0.10	0.40	0.40	0.20	1.00000000	0.43452381	0.56547619
41	4	1.0	0.35	0.35	0.30	1.00	0.00	0.00	1.00000000	0.51587302	0.48412698
42	4	1.0	0.35	0.35	0.30	0.50	0.40	0.10	1.00000000	0.44166667	0.55833333
43	4	1.0	0.35	0.35	0.30	0.50	0.30	0.20	1.00000000	0.41666667	0.58333333
44	4	1.0	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.49702381	0.50297619
45	4	1.0	0.35	0.35	0.30	0.40	0.40	0.20	1.00000000	0.38194444	0.61805556
46	4	1.0	0.40	0.35	0.25	1.00	0.00	0.00	1.00000000	0.59166667	0.40833333
47	4	1.0	0.40	0.35	0.25	0.50	0.40	0.10	1.00000000	0.41666667	0.58333333
48	4	1.0	0.40	0.35	0.25	0.50	0.30	0.20	1.00000000	0.54166667	0.45833333
49	4	1.0	0.40	0.35	0.25	0.60	0.20	0.20	1.00000000	0.48518519	0.51481481
50	4	1.0	0.40	0.35	0.25	0.40	0.40	0.20	1.00000000	0.50000000	0.50000000
51	5	1.0	0.40	0.40	0.20	1.00	0.00	0.00	1.00000000	0.38095238	0.61904762
52	5	1.0	0.40	0.40	0.20	0.50	0.40	0.10	0.96428571	0.48666667	0.54904762
53	5	1.0	0.40	0.40	0.20	0.50	0.30	0.20	1.00000000	0.42261905	0.57738095
54	5	1.0	0.40	0.40	0.20	0.60	0.20	0.20	1.00000000	0.41666667	0.58333333
55	5	1.0	0.40	0.40	0.20	0.40	0.40	0.20	0.96428571	0.40740741	0.62830688
56	5	1.0	0.50	0.30	0.20	1.00	0.00	0.00	0.96428571	0.45833333	0.57738095
57	5	1.0	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.38194444	0.61805556
58	5	1.0	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.42962963	0.57037037
59	5	1.0	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.42962963	0.57037037
60	5	1.0	0.50	0.30	0.20	0.40	0.40	0.20	1.00000000	0.43452381	0.56547619
61	5	1.0	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.49479167	0.54092262
62	5	1.0	0.45	0.45	0.10	0.50	0.40	0.10	0.96428571	0.41369048	0.62202381
63	5	1.0	0.45	0.45	0.10	0.50	0.30	0.20	0.96428571	0.41369048	0.62202381
64	5	1.0	0.45	0.45	0.10	0.60	0.20	0.20	0.92857143	0.43981481	0.63161376
65	5	1.0	0.45	0.45	0.10	0.40	0.40	0.20	0.96428571	0.41369048	0.62202381
66	5	1.0	0.35	0.35	0.30	1.00	0.00	0.00	1.00000000	0.43333333	0.56666667
67	5	1.0	0.35	0.35	0.30	0.50	0.40	0.10	1.00000000	0.44166667	0.55833333
68	5	1.0	0.35	0.35	0.30	0.50	0.30	0.20	1.00000000	0.41666667	0.58333333

Continua na próxima página ...

... continuação da página anterior

n°	tp	smín	F <sub>S</sub>	F <sub>O</sub>	F <sub>N</sub>	F <sub>NM</sub>	F <sub>NC</sub>	F <sub>PR</sub>	Dispersão	Diversificação	Diferença
69	5	1.0	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.45000000	0.55000000
70	5	1.0	0.35	0.35	0.30	0.40	0.40	0.20	1.00000000	0.42962963	0.57037037
71	5	1.0	0.40	0.35	0.25	1.00	0.00	0.00	1.00000000	0.43333333	0.56666667
72	5	1.0	0.40	0.35	0.25	0.50	0.40	0.10	1.00000000	0.40000000	0.60000000
73	5	1.0	0.40	0.35	0.25	0.50	0.30	0.20	1.00000000	0.43452381	0.56547619
74	5	1.0	0.40	0.35	0.25	0.60	0.20	0.20	1.00000000	0.41041667	0.58958333
75	5	1.0	0.40	0.35	0.25	0.40	0.40	0.20	1.00000000	0.46212121	0.53787879
76	3	0.8	0.40	0.40	0.20	1.00	0.00	0.00	0.96428571	0.76666667	0.26904762
77	3	0.8	0.40	0.40	0.20	0.50	0.40	0.10	0.89285714	0.58333333	0.52380952
78	3	0.8	0.40	0.40	0.20	0.50	0.30	0.20	0.96428571	0.68333333	0.35238095
79	3	0.8	0.40	0.40	0.20	0.60	0.20	0.20	1.00000000	0.74545455	0.25454545
80	3	0.8	0.40	0.40	0.20	0.40	0.40	0.20	0.92857143	0.51250000	0.55892857
81	3	0.8	0.50	0.30	0.20	1.00	0.00	0.00	0.96428571	0.76428571	0.27142857
82	3	0.8	0.50	0.30	0.20	0.50	0.40	0.10	0.92857143	0.66923077	0.40219780
83	3	0.8	0.50	0.30	0.20	0.50	0.30	0.20	0.92857143	0.58636364	0.48506494
84	3	0.8	0.50	0.30	0.20	0.60	0.20	0.20	0.96428571	0.75000000	0.28571429
85	3	0.8	0.50	0.30	0.20	0.40	0.40	0.20	0.92857143	0.60606061	0.46536797
86	3	0.8	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.70238095	0.33333333
87	3	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.89285714	0.64705882	0.46008403
88	3	0.8	0.45	0.45	0.10	0.50	0.30	0.20	0.89285714	0.67647059	0.43067227
89	3	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.89285714	0.61764706	0.48949580
90	3	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.89285714	0.58823529	0.51890756
91	3	0.8	0.35	0.35	0.30	1.00	0.00	0.00	0.96428571	0.63939394	0.39632035
92	3	0.8	0.35	0.35	0.30	0.50	0.40	0.10	0.96428571	0.60846561	0.42724868
93	3	0.8	0.35	0.35	0.30	0.50	0.30	0.20	0.96428571	0.67361111	0.36210317
94	3	0.8	0.35	0.35	0.30	0.60	0.20	0.20	0.96428571	0.58571429	0.45000000
95	3	0.8	0.35	0.35	0.30	0.40	0.40	0.20	0.92857143	0.62179487	0.44963370
96	3	0.8	0.40	0.35	0.25	1.00	0.00	0.00	0.96428571	0.83333333	0.20238095
97	3	0.8	0.40	0.35	0.25	0.50	0.40	0.10	0.96428571	0.74358974	0.29212454
98	3	0.8	0.40	0.35	0.25	0.50	0.30	0.20	0.96428571	0.65151515	0.38419913
99	3	0.8	0.40	0.35	0.25	0.60	0.20	0.20	0.96428571	0.63194444	0.40376984
100	3	0.8	0.40	0.35	0.25	0.40	0.40	0.20	0.92857143	0.58611111	0.48531746
101	4	0.8	0.40	0.40	0.20	1.00	0.00	0.00	1.00000000	0.65740741	0.34259259
102	4	0.8	0.40	0.40	0.20	0.50	0.40	0.10	1.00000000	0.40136054	0.59863946
103	4	0.8	0.40	0.40	0.20	0.50	0.30	0.20	1.00000000	0.46527778	0.53472222
104	4	0.8	0.40	0.40	0.20	0.60	0.20	0.20	1.00000000	0.49074074	0.50925926
105	4	0.8	0.40	0.40	0.20	0.40	0.40	0.20	1.00000000	0.47278912	0.52721088
106	4	0.8	0.50	0.30	0.20	1.00	0.00	0.00	1.00000000	0.50000000	0.50000000
107	4	0.8	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.49074074	0.50925926
108	4	0.8	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.41041667	0.58958333
109	4	0.8	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.47278912	0.52721088
110	4	0.8	0.50	0.30	0.20	0.40	0.40	0.20	1.00000000	0.43452381	0.56547619
111	4	0.8	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.47619048	0.55952381
112	4	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.96428571	0.39880952	0.63690476
113	4	0.8	0.45	0.45	0.10	0.50	0.30	0.20	1.00000000	0.38194444	0.61805556
114	4	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.92857143	0.43055556	0.64087302
115	4	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.96428571	0.41369048	0.62202381
116	4	0.8	0.35	0.35	0.30	1.00	0.00	0.00	1.00000000	0.48958333	0.51041667
117	4	0.8	0.35	0.35	0.30	0.50	0.40	0.10	1.00000000	0.44166667	0.55833333
118	4	0.8	0.35	0.35	0.30	0.50	0.30	0.20	1.00000000	0.41666667	0.58333333
119	4	0.8	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.49404762	0.50592381
120	4	0.8	0.35	0.35	0.30	0.40	0.40	0.20	1.00000000	0.41666667	0.58333333
121	4	0.8	0.40	0.35	0.25	1.00	0.00	0.00	1.00000000	0.51818182	0.48181818
122	4	0.8	0.40	0.35	0.25	0.50	0.40	0.10	1.00000000	0.38666667	0.61333333
123	4	0.8	0.40	0.35	0.25	0.50	0.30	0.20	1.00000000	0.40136054	0.59863946
124	4	0.8	0.40	0.35	0.25	0.60	0.20	0.20	1.00000000	0.49702381	0.50297619
125	4	0.8	0.40	0.35	0.25	0.40	0.40	0.20	1.00000000	0.40136054	0.59863946
126	5	0.8	0.40	0.40	0.20	1.00	0.00	0.00	1.00000000	0.33333333	0.66666667
127	5	0.8	0.40	0.40	0.20	0.50	0.40	0.10	1.00000000	0.41666667	0.58333333
128	5	0.8	0.40	0.40	0.20	0.50	0.30	0.20	1.00000000	0.41666667	0.58333333
129	5	0.8	0.40	0.40	0.20	0.60	0.20	0.20	1.00000000	0.42261905	0.57738095
130	5	0.8	0.40	0.40	0.20	0.40	0.40	0.20	0.96428571	0.46666667	0.56904762
131	5	0.8	0.50	0.30	0.20	1.00	0.00	0.00	1.00000000	0.41666667	0.58333333
132	5	0.8	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.38666667	0.61333333
133	5	0.8	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.42261905	0.57738095
134	5	0.8	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.42261905	0.57738095
135	5	0.8	0.50	0.30	0.20	0.40	0.40	0.20	1.00000000	0.43452381	0.56547619
136	5	0.8	0.45	0.45	0.10	1.00	0.00	0.00	1.00000000	0.49074074	0.50925926
137	5	0.8	0.45	0.45	0.10	0.50	0.40	0.10	0.92857143	0.41145833	0.65997024
138	5	0.8	0.45	0.45	0.10	0.50	0.30	0.20	1.00000000	0.38194444	0.61805556
139	5	0.8	0.45	0.45	0.10	0.60	0.20	0.20	0.96428571	0.39880952	0.63690476
140	5	0.8	0.45	0.45	0.10	0.40	0.40	0.20	0.96428571	0.37962963	0.65608466
141	5	0.8	0.35	0.35	0.30	1.00	0.00	0.00	1.00000000	0.37222222	0.62777778
142	5	0.8	0.35	0.35	0.30	0.50	0.40	0.10	1.00000000	0.40136054	0.59863946
143	5	0.8	0.35	0.35	0.30	0.50	0.30	0.20	1.00000000	0.38666667	0.61333333
144	5	0.8	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.42261905	0.57738095
145	5	0.8	0.35	0.35	0.30	0.40	0.40	0.20	1.00000000	0.40740741	0.59259259
146	5	0.8	0.40	0.35	0.25	1.00	0.00	0.00	1.00000000	0.42261905	0.57738095
147	5	0.8	0.40	0.35	0.25	0.50	0.40	0.10	1.00000000	0.42962963	0.57037037
148	5	0.8	0.40	0.35	0.25	0.50	0.30	0.20	1.00000000	0.49404762	0.50592381
149	5	0.8	0.40	0.35	0.25	0.60	0.20	0.20	1.00000000	0.40740741	0.59259259
150	5	0.8	0.40	0.35	0.25	0.40	0.40	0.20	1.00000000	0.34166667	0.65833333
151	3	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.92857143	0.78125000	0.29017857
152	3	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.89285714	0.64705882	0.46008403
153	3	1.2	0.40	0.40	0.20	0.50	0.30	0.20	0.92857143	0.75000000	0.32142857
154	3	1.2	0.40	0.40	0.20	0.60	0.20	0.20	0.92857143	0.75000000	0.32142857
155	3	1.2	0.40	0.40	0.20	0.40	0.40	0.20	0.89285714	0.70588235	0.40126050
156	3	1.2	0.50	0.30	0.20	1.00	0.00	0.00	0.96428571	0.76428571	0.27142857
157	3	1.2	0.50	0.30	0.20	0.50	0.40	0.10	0.89285714	0.73529412	0.37184874
158	3	1.2	0.50	0.30	0.20	0.50	0.30	0.20	0.92857143	0.78125000	0.29017857
159	3	1.2	0.50	0.30	0.20	0.60	0.20	0.20	0.96428571	0.83333333	0.20238095

Continua na próxima página ...

... continuação da página anterior											
$n^{\circ}$	$tp$	$sm_{in}$	$F_S$	$F_O$	$F_N$	$F_{NM}$	$F_{NC}$	$F_{PR}$	Dispersão	Diversificação	Diferença
160	3	1.2	0.50	0.30	0.20	0.40	0.40	0.20	0.92857143	0.63333333	0.43809524
161	3	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.92857143	0.71875000	0.35267857
162	3	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.89285714	0.64705882	0.46008403
163	3	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.89285714	0.67647059	0.43067227
164	3	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.89285714	0.70588235	0.40126050
165	3	1.2	0.45	0.45	0.10	0.40	0.40	0.20	0.89285714	0.67647059	0.43067227
166	3	1.2	0.35	0.35	0.30	1.00	0.00	0.00	0.96428571	0.75000000	0.28571429
167	3	1.2	0.35	0.35	0.30	0.50	0.40	0.10	0.92857143	0.75000000	0.32142857
168	3	1.2	0.35	0.35	0.30	0.50	0.30	0.20	0.96428571	0.76666667	0.26904762
169	3	1.2	0.35	0.35	0.30	0.60	0.20	0.20	0.96428571	0.73333333	0.30238095
170	3	1.2	0.35	0.35	0.30	0.40	0.40	0.20	0.92857143	0.75000000	0.32142857
171	3	1.2	0.40	0.35	0.25	1.00	0.00	0.00	0.96428571	0.86666667	0.16904762
172	3	1.2	0.40	0.35	0.25	0.50	0.40	0.10	0.92857143	0.75000000	0.32142857
173	3	1.2	0.40	0.35	0.25	0.50	0.30	0.20	0.96428571	0.71153846	0.32417582
174	3	1.2	0.40	0.35	0.25	0.60	0.20	0.20	0.96428571	0.76666667	0.26904762
175	3	1.2	0.40	0.35	0.25	0.40	0.40	0.20	0.92857143	0.68750000	0.38392857
176	4	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.96428571	0.58333333	0.45238095
177	4	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.92857143	0.44242424	0.62900433
178	4	1.2	0.40	0.40	0.20	0.50	0.30	0.20	1.00000000	0.47278912	0.52721088
179	4	1.2	0.40	0.40	0.20	0.60	0.20	0.20	1.00000000	0.49074074	0.50923926
180	4	1.2	0.40	0.40	0.20	0.40	0.40	0.20	1.00000000	0.49702381	0.50297619
181	4	1.2	0.50	0.30	0.20	1.00	0.00	0.00	1.00000000	0.49166667	0.50833333
182	4	1.2	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.59722222	0.40277778
183	4	1.2	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.48148148	0.51851852
184	4	1.2	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.47916667	0.52083333
185	4	1.2	0.50	0.30	0.20	0.40	0.40	0.20	1.00000000	0.50333333	0.49666667
186	4	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.48666667	0.54904762
187	4	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.92857143	0.42328042	0.64814815
188	4	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.96428571	0.41369048	0.62202381
189	4	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.92857143	0.43981481	0.63161376
190	4	1.2	0.45	0.45	0.10	0.40	0.40	0.20	1.00000000	0.42962963	0.57037037
191	4	1.2	0.35	0.35	0.30	1.00	0.00	0.00	1.00000000	0.47222222	0.52777778
192	4	1.2	0.35	0.35	0.30	0.50	0.40	0.10	1.00000000	0.50757576	0.49242424
193	4	1.2	0.35	0.35	0.30	0.50	0.30	0.20	1.00000000	0.49166667	0.50833333
194	4	1.2	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.50000000	0.50000000
195	4	1.2	0.35	0.35	0.30	0.40	0.40	0.20	1.00000000	0.42962963	0.57037037
196	4	1.2	0.40	0.35	0.25	1.00	0.00	0.00	1.00000000	0.62698413	0.37301587
197	4	1.2	0.40	0.35	0.25	0.50	0.40	0.10	1.00000000	0.49166667	0.50833333
198	4	1.2	0.40	0.35	0.25	0.50	0.30	0.20	1.00000000	0.54166667	0.45833333
199	4	1.2	0.40	0.35	0.25	0.60	0.20	0.20	1.00000000	0.49166667	0.50833333
200	4	1.2	0.40	0.35	0.25	0.40	0.40	0.20	1.00000000	0.55555556	0.44444444
201	5	1.2	0.40	0.40	0.20	1.00	0.00	0.00	0.96428571	0.48484848	0.55086580
202	5	1.2	0.40	0.40	0.20	0.50	0.40	0.10	0.96428571	0.41111111	0.62460317
203	5	1.2	0.40	0.40	0.20	0.50	0.30	0.20	0.96428571	0.43229167	0.60342262
204	5	1.2	0.40	0.40	0.20	0.60	0.20	0.20	0.96428571	0.42592593	0.60978836
205	5	1.2	0.40	0.40	0.20	0.40	0.40	0.20	0.96428571	0.46527778	0.57043651
206	5	1.2	0.50	0.30	0.20	1.00	0.00	0.00	1.00000000	0.44166667	0.55833333
207	5	1.2	0.50	0.30	0.20	0.50	0.40	0.10	1.00000000	0.43333333	0.56666667
208	5	1.2	0.50	0.30	0.20	0.50	0.30	0.20	1.00000000	0.44166667	0.55833333
209	5	1.2	0.50	0.30	0.20	0.60	0.20	0.20	1.00000000	0.41666667	0.58333333
210	5	1.2	0.50	0.30	0.20	0.40	0.40	0.20	1.00000000	0.43452381	0.56547619
211	5	1.2	0.45	0.45	0.10	1.00	0.00	0.00	0.96428571	0.43229167	0.60342262
212	5	1.2	0.45	0.45	0.10	0.50	0.40	0.10	0.92857143	0.46153846	0.60989011
213	5	1.2	0.45	0.45	0.10	0.50	0.30	0.20	0.96428571	0.41369048	0.62202381
214	5	1.2	0.45	0.45	0.10	0.60	0.20	0.20	0.96428571	0.50000000	0.53571429
215	5	1.2	0.45	0.45	0.10	0.40	0.40	0.20	0.96428571	0.45833333	0.57738095
216	5	1.2	0.35	0.35	0.30	1.00	0.00	0.00	1.00000000	0.40625000	0.59375000
217	5	1.2	0.35	0.35	0.30	0.50	0.40	0.10	1.00000000	0.44166667	0.55833333
218	5	1.2	0.35	0.35	0.30	0.50	0.30	0.20	1.00000000	0.45333333	0.54666667
219	5	1.2	0.35	0.35	0.30	0.60	0.20	0.20	1.00000000	0.40136054	0.59863946
220	5	1.2	0.35	0.35	0.30	0.40	0.40	0.20	1.00000000	0.41666667	0.58333333
221	5	1.2	0.40	0.35	0.25	1.00	0.00	0.00	1.00000000	0.45454545	0.54545455
222	5	1.2	0.40	0.35	0.25	0.50	0.40	0.10	1.00000000	0.46212121	0.53787879
223	5	1.2	0.40	0.35	0.25	0.50	0.30	0.20	1.00000000	0.50757576	0.49242424
224	5	1.2	0.40	0.35	0.25	0.60	0.20	0.20	1.00000000	0.46212121	0.53787879
225	5	1.2	0.40	0.35	0.25	0.40	0.40	0.20	1.00000000	0.38095238	0.61904762

## APÊNDICE C

### AGRUPAMENTO E ORDENAÇÃO

Este apêndice apresenta os grupos identificados pela instância 12 da fase de agrupamento respectivamente nos sistemas JHotDraw, Tomcat e HSQLDB. Cada grupo teve a sua pontuação calculada com a aplicação respectiva das medidas de ordenação GSRank, GSI-Rank, GFRank e GFIRank.

Para representar os grupos obtidos pela fase de agrupamento, foram geradas duas tabelas distintas contendo respectivamente para cada sistema: os métodos com os identificadores e os grupos com os identificadores dos métodos que compõem o grupo.

Na Tabela C.1 são apresentados os métodos do sistema JHotDraw com o respectivo *fan-in*. Devido a limitação de espaço, foram listados somente os métodos com *fan-in* a partir de 5.

Tabela C.1: Lista de métodos do JHotDraw

[Id]	Método(Parâmetros)	<i>fan-in</i>
[22]	framework.DrawingView applet.DrawApplet.view()	14
[24]	void applet.DrawApplet.toolDone()	16
[25]	void applet.DrawApplet.figureSelectionChanged(framework.DrawingView)	5
[51]	void application.DrawApplication.newWindow(framework.Drawing)	6
[53]	void application.DrawApplication.open()	7
[70]	void application.DrawApplication.createTools(JToolBar)	6
[75]	standard.ToolButton application.DrawApplication.createToolButton(String,String,framework.Tool)	6
[80]	framework.Drawing application.DrawApplication.createDrawing()	5
[93]	framework.Tool application.DrawApplication.tool()	6
[94]	framework.DrawingView application.DrawApplication.view()	16
[97]	void application.DrawApplication.toolDone()	18
[98]	void application.DrawApplication.figureSelectionChanged(framework.DrawingView)	5
[106]	void application.DrawApplication.showStatus(String)	6
[167]	void contrib.ComponentFigure.basicDisplayBox(Point,Point)	5
[169]	Rectangle contrib.ComponentFigure.displayBox()	49
[173]	void contrib.ComponentFigure.draw(Graphics)	5
[175]	void contrib.CompositeFigureCreationTool.mouseDown(event.MouseEvent,int,int)	18
[186]	void contrib.CTXCommandMenu.addItem(JMenuItem)	5
[201]	void contrib.CustomSelectionTool.mouseDown(event.MouseEvent,int,int)	17
[202]	void contrib.CustomSelectionTool.mouseDrag(event.MouseEvent,int,int)	10
[203]	void contrib.CustomSelectionTool.mouseUp(event.MouseEvent,int,int)	10
[255]	void contrib.DiamondFigure.draw(Graphics)	5
[257]	boolean contrib.DiamondFigure.containsPoint(int,int)	16
[259]	framework.Connector contrib.DiamondFigure.connectorAt(int,int)	6
[293]	void contrib.dnd.DragNDropTool.activate()	9
[294]	void contrib.dnd.DragNDropTool.deactivate()	14
[298]	void contrib.dnd.DragNDropTool.mouseUp(event.MouseEvent,int,int)	9
[299]	void contrib.dnd.DragNDropTool.mouseDown(event.MouseEvent,int,int)	15
[300]	void contrib.dnd.DragNDropTool.mouseDrag(event.MouseEvent,int,int)	9
[314]	Object contrib.GraphicalCompositeFigure.clone()	8
[315]	Rectangle contrib.GraphicalCompositeFigure.displayBox()	50
[316]	void contrib.GraphicalCompositeFigure.basicDisplayBox(Point,Point)	5
[319]	void contrib.GraphicalCompositeFigure.draw(Graphics)	5
[331]	void contrib.GraphicalCompositeFigure.figureRequestRemove(framework.FigureChangeEvent)	6
[332]	void contrib.GraphicalCompositeFigure.read(util.StorableInput)	7
[333]	void contrib.GraphicalCompositeFigure.write(util.StorableOutput)	7
[338]	void contrib.html.AbstractContentProducer.write(util.StorableOutput)	7
[339]	void contrib.html.AbstractContentProducer.read(util.StorableInput)	7
[343]	void contrib.html.AttributeFigureContentProducer.write(util.StorableOutput)	7
[344]	void contrib.html.AttributeFigureContentProducer.read(util.StorableInput)	7
[350]	void contrib.html.ColorContentProducer.write(util.StorableOutput)	7
[351]	void contrib.html.ColorContentProducer.read(util.StorableInput)	7
[419]	void contrib.html.FigureDataContentProducer.write(util.StorableOutput)	7
[420]	void contrib.html.FigureDataContentProducer.read(util.StorableInput)	7
[438]	Object contrib.html.HTMLTextAreaFigure.clone()	8
[439]	void contrib.html.HTMLTextAreaFigure.basicDisplayBox(Point,Point)	7
[441]	boolean contrib.html.HTMLTextAreaFigure.containsPoint(int,int)	16
[442]	void contrib.html.HTMLTextAreaFigure.moveBy(int,int)	12
[444]	void contrib.html.HTMLTextAreaFigure.markSizeDirty()	7
[445]	void contrib.html.HTMLTextAreaFigure.markTextDirty()	5
[446]	void contrib.html.HTMLTextAreaFigure.markFontDirty()	5
[447]	void contrib.html.HTMLTextAreaFigure.draw(Graphics)	5
[469]	void contrib.html.HTMLTextAreaFigure.markImageDirty()	5
[471]	void contrib.html.HTMLTextAreaFigure.read(util.StorableInput)	13
[472]	void contrib.html.HTMLTextAreaFigure.write(util.StorableOutput)	13
[487]	void contrib.html.HTMLTextAreaFigure.figureRemoved(framework.FigureChangeEvent)	5
[488]	void contrib.html.HTMLTextAreaFigure.figureRequestRemove(framework.FigureChangeEvent)	6
[489]	void contrib.html.HTMLTextAreaFigure.figureRequestUpdate(framework.FigureChangeEvent)	6
[545]	void contrib.html.URLContentProducer.write(util.StorableOutput)	7

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
546]	void contrib.html.URLContentProducer.read(util.StorableInput)	7
607]	void contrib.MDIDesktopPane.checkDesktopSize()	7
620]	void contrib.MDI_DrawApplication.createTools(JToolBar)	6
622]	void contrib.MDI_DrawApplication.newWindow(framework.Drawing)	6
655]	void contrib.NestedCreationTool.mouseDown(event.MouseEvent,int,int)	18
657]	void contrib.NestedCreationTool.mouseUp(event.MouseEvent,int,int)	9
664]	Rectangle contrib.PolygonFigure.displayBox()	49
667]	void contrib.PolygonFigure.basicDisplayBox(Point,Point)	5
671]	Point contrib.PolygonFigure.center()	9
673]	int contrib.PolygonFigure.pointCount()	12
677]	boolean contrib.PolygonFigure.containsPoint(int,int)	16
678]	framework.Connector contrib.PolygonFigure.connectorAt(int,int)	6
684]	void contrib.PolygonFigure.smoothPoints()	5
690]	void contrib.PolygonFigure.write(util.StorableOutput)	12
691]	void contrib.PolygonFigure.read(util.StorableInput)	12
714]	boolean contrib.PolygonScaleHandle\$UndoActivity.undo()	24
715]	boolean contrib.PolygonScaleHandle\$UndoActivity.redo()	7
720]	void contrib.PolygonTool.activate()	9
721]	void contrib.PolygonTool.deactivate()	14
723]	void contrib.PolygonTool.mouseDown(event.MouseEvent,int,int)	15
725]	void contrib.PolygonTool.mouseDrag(event.MouseEvent,int,int)	9
726]	void contrib.PolygonTool.mouseUp(event.MouseEvent,int,int)	9
744]	void contrib.SplitConnectionTool.mouseDown(event.MouseEvent,int,int)	15
745]	void contrib.SplitConnectionTool.mouseUp(event.MouseEvent,int,int)	9
747]	void contrib.SplitConnectionTool.mouseDrag(event.MouseEvent,int,int)	9
748]	void contrib.SplitConnectionTool.deactivate()	15
793]	void contrib.TextAreaFigure.markTextDirty()	5
796]	void contrib.TextAreaFigure.markSizeDirty()	7
802]	void contrib.TextAreaFigure.basicDisplayBox(Point,Point)	7
804]	Rectangle contrib.TextAreaFigure.displayBox()	54
805]	void contrib.TextAreaFigure.moveBy(int,int)	12
808]	void contrib.TextAreaFigure.draw(Graphics)	5
815]	void contrib.TextAreaFigure.write(util.StorableOutput)	13
816]	void contrib.TextAreaFigure.read(util.StorableInput)	13
823]	void contrib.TextAreaFigure.figureRemoved(framework.FigureChangeEvent)	5
824]	void contrib.TextAreaFigure.figureRequestRemove(framework.FigureChangeEvent)	6
825]	void contrib.TextAreaFigure.figureRequestUpdate(framework.FigureChangeEvent)	6
830]	void contrib.TextAreaFigure.markFontDirty()	5
835]	void contrib.TextAreaTool.mouseDown(event.MouseEvent,int,int)	18
836]	void contrib.TextAreaTool.mouseDrag(event.MouseEvent,int,int)	10
837]	void contrib.TextAreaTool.mouseUp(event.MouseEvent,int,int)	9
838]	void contrib.TextAreaTool.deactivate()	14
839]	void contrib.TextAreaTool.activate()	11
851]	boolean contrib.TextAreaTool\$UndoActivity.undo()	24
852]	boolean contrib.TextAreaTool\$UndoActivity.redo()	7
861]	framework.HandleEnumeration contrib.TriangleFigure.handles()	5
864]	void contrib.TriangleFigure.draw(Graphics)	5
866]	boolean contrib.TriangleFigure.containsPoint(int,int)	16
867]	Point contrib.TriangleFigure.center()	7
869]	Object contrib.TriangleFigure.clone()	8
871]	void contrib.TriangleFigure.write(util.StorableOutput)	13
872]	void contrib.TriangleFigure.read(util.StorableInput)	13
882]	boolean contrib.TriangleRotationHandle\$UndoActivity.undo()	24
883]	boolean contrib.TriangleRotationHandle\$UndoActivity.redo()	7
893]	void contrib.zoom.AreaTracker.mouseDown(event.MouseEvent,int,int)	15
894]	void contrib.zoom.AreaTracker.mouseDrag(event.MouseEvent,int,int)	9
895]	void contrib.zoom.AreaTracker.mouseUp(event.MouseEvent,int,int)	10
960]	void contrib.zoom.ZoomAreaTracker.mouseUp(event.MouseEvent,int,int)	10
962]	void contrib.zoom.ZoomCommand.execute()	24
971]	void contrib.zoom.ZoomDrawingView.forceRedraw()	8
976]	boolean contrib.zoom.ZoomDrawingView.hasZoomSupport()	5
1004]	void contrib.zoom.ZoomTool.mouseDown(event.MouseEvent,int,int)	15
1005]	void contrib.zoom.ZoomTool.mouseDrag(event.MouseEvent,int,int)	9
1006]	void contrib.zoom.ZoomTool.mouseUp(event.MouseEvent,int,int)	9
1033]	void figures.AttributeFigure.draw(Graphics)	5
1047]	void figures.AttributeFigure.write(util.StorableOutput)	14
1048]	void figures.AttributeFigure.read(util.StorableInput)	14
1056]	Rectangle figures.Border.displayBox()	50
1057]	void figures.BorderDecorator.figureInvalidated(framework.FigureChangeEvent)	5
1058]	Insets figures.BorderDecorator.connectionInsets()	5
1060]	void figures.BorderTool.mouseDown(event.MouseEvent,int,int)	16
1065]	boolean figures.BorderTool\$UndoActivity.undo()	24
1066]	boolean figures.BorderTool\$UndoActivity.redo()	7
1072]	void figures.ConnectedTextTool.mouseDown(event.MouseEvent,int,int)	19
1076]	void figures.ConnectedTextTool.activate()	12
1080]	boolean figures.ConnectedTextTool\$UndoActivity.undo()	25
1081]	boolean figures.ConnectedTextTool\$UndoActivity.redo()	8
1085]	boolean figures.ConnectedTextTool\$DeleteUndoActivity.undo()	25
1086]	boolean figures.ConnectedTextTool\$DeleteUndoActivity.redo()	8
1090]	void figures.ElbowConnection.updateConnection()	12
1092]	framework.HandleEnumeration figures.ElbowConnection.handles()	5
1109]	void figures.EllipseFigure.basicDisplayBox(Point,Point)	6
1110]	Rectangle figures.EllipseFigure.displayBox()	51
1115]	framework.Connector figures.EllipseFigure.connectorAt(int,int)	6
1116]	void figures.EllipseFigure.write(util.StorableOutput)	12
1117]	void figures.EllipseFigure.read(util.StorableInput)	12
1134]	boolean figures.FontSizeHandle\$UndoActivity.undo()	24
1135]	boolean figures.FontSizeHandle\$UndoActivity.redo()	7
1143]	void figures.GroupCommand.execute()	24
1147]	boolean figures.GroupCommand\$UndoActivity.undo()	24
1148]	boolean figures.GroupCommand\$UndoActivity.redo()	7

Continua na próxima página ...

... continuação da página anterior	
[Id] Método(Parâmetros)	fan-in
1151 boolean figures.GroupFigure.canConnect()	9
1152 Rectangle figures.GroupFigure.displayBox()	48
1153 void figures.GroupFigure.basicDisplayBox(Point,Point)	5
1161 void figures.ImageFigure.basicDisplayBox(Point,Point)	6
1163 Rectangle figures.ImageFigure.displayBox()	48
1165 void figures.ImageFigure.draw(Graphics)	5
1168 void figures.ImageFigure.release()	6
1169 void figures.ImageFigure.write(util.StorableOutput)	12
1170 void figures.ImageFigure.read(util.StorableInput)	12
1173 void figures.InsertImageCommand.execute()	24
1178 boolean figures.InsertImageCommand\$UndoActivity.undo()	24
1179 boolean figures.InsertImageCommand\$UndoActivity.redo()	7
1182 boolean figures.LineConnection.canConnect()	9
1184 void figures.LineConnection.connectStart/framework.Connector)	9
1185 void figures.LineConnection.connectEnd/framework.Connector)	9
1191 framework.Figure figures.LineConnection.startFigure()	7
1192 framework.Figure figures.LineConnection.endFigure()	5
1198 void figures.LineConnection.startPoint(int,int)	5
1199 void figures.LineConnection.endPoint(int,int)	7
1202 framework.HandleEnumeration figures.LineConnection.handles()	5
1206 void figures.LineConnection.updateConnection()	12
1209 void figures.LineConnection.figureRemoved/framework.FigureChangeEvent)	5
1210 void figures.LineConnection.figureRequestRemove/framework.FigureChangeEvent)	6
1212 void figures.LineConnection.figureRequestUpdate/framework.FigureChangeEvent)	6
1213 void figures.LineConnection.release()	6
1214 void figures.LineConnection.write(util.StorableOutput)	6
1215 void figures.LineConnection.read(util.StorableInput)	6
1226 void figures.LineFigure.basicDisplayBox(Point,Point)	5
1231 void figures.NullFigure.basicDisplayBox(Point,Point)	5
1232 Rectangle figures.NullFigure.displayBox()	48
1236 framework.FigureEnumeration figures.NullFigure.figures()	9
1238 Object figures.NullFigure.clone()	8
1239 boolean figures.NullFigure.includes/framework.Figure)	13
1241 void figures.NullFigure.release()	6
1242 void figures.NullFigure.invalidate()	9
1258 Rectangle figures.PolyLineFigure.displayBox()	49
1260 framework.HandleEnumeration figures.PolyLineFigure.handles()	5
1261 void figures.PolyLineFigure.basicDisplayBox(Point,Point)	5
1264 int figures.PolyLineFigure.pointCount()	5
1270 Point figures.PolyLineFigure.pointAt(int)	15
1272 framework.Connector figures.PolyLineFigure.connectorAt(int,int)	6
1279 boolean figures.PolyLineFigure.containsPoint(int,int)	16
1286 void figures.PolyLineFigure.write(util.StorableOutput)	6
1287 void figures.PolyLineFigure.read(util.StorableInput)	6
1291 Rectangle figures.PolyLineFigure.invalidateRectangle(Rectangle)	6
1299 boolean figures.PolyLineHandle\$UndoActivity.undo()	24
1300 boolean figures.PolyLineHandle\$UndoActivity.redo()	7
1316 boolean figures.RadiusHandle\$UndoActivity.undo()	24
1317 boolean figures.RadiusHandle\$UndoActivity.redo()	7
1323 void figures.RectangleFigure.basicDisplayBox(Point,Point)	6
1324 framework.HandleEnumeration figures.RectangleFigure.handles()	5
1325 Rectangle figures.RectangleFigure.displayBox()	54
1329 void figures.RectangleFigure.write(util.StorableOutput)	13
1330 void figures.RectangleFigure.read(util.StorableInput)	13
1333 void figures.RoundRectangleFigure.basicDisplayBox(Point,Point)	6
1337 Rectangle figures.RoundRectangleFigure.displayBox()	53
1342 framework.Connector figures.RoundRectangleFigure.connectorAt(int,int)	6
1343 void figures.RoundRectangleFigure.write(util.StorableOutput)	12
1344 void figures.RoundRectangleFigure.read(util.StorableInput)	12
1346 void figures.ScribbleTool.activate()	9
1347 void figures.ScribbleTool.deactivate()	14
1349 void figures.ScribbleTool.mouseDown(event.MouseEvent,int,int)	15
1350 void figures.ScribbleTool.mouseDrag(event.MouseEvent,int,int)	9
1351 void figures.ScribbleTool.mouseUp(event.MouseEvent,int,int)	9
1361 void figures.TextFigure.moveBy(int,int)	12
1363 void figures.TextFigure.basicDisplayBox(Point,Point)	5
1364 Rectangle figures.TextFigure.displayBox()	54
1371 void figures.TextFigure.changed()	37
1385 void figures.TextFigure.write(util.StorableOutput)	12
1386 void figures.TextFigure.read(util.StorableInput)	12
1390 void figures.TextFigure.figureRemoved/framework.FigureChangeEvent)	5
1391 void figures.TextFigure.figureRequestRemove/framework.FigureChangeEvent)	6
1393 void figures.TextFigure.figureRequestUpdate/framework.FigureChangeEvent)	6
1395 void figures.TextFigure.release()	6
1407 void figures.TextTool.mouseDown(event.MouseEvent,int,int)	19
1408 void figures.TextTool.mouseDrag(event.MouseEvent,int,int)	10
1409 void figures.TextTool.mouseUp(event.MouseEvent,int,int)	9
1410 void figures.TextTool.deactivate()	14
1411 void figures.TextTool.activate()	12
1428 boolean figures.TextTool\$UndoActivity.undo()	25
1429 boolean figures.TextTool\$UndoActivity.redo()	8
1437 void figures.UngroupCommand.execute()	24
1441 boolean figures.UngroupCommand\$UndoActivity.undo()	24
1442 boolean figures.UngroupCommand\$UndoActivity.redo()	7
1444 void framework.ConnectionFigure.connectStart/framework.Connector)	9
1445 void framework.ConnectionFigure.connectEnd/framework.Connector)	9
1446 void framework.ConnectionFigure.updateConnection()	12
1453 void framework.ConnectionFigure.startPoint(int,int)	5
1454 void framework.ConnectionFigure.endPoint(int,int)	7
1462 framework.Figure framework.ConnectionFigure.startFigure()	7
1463 framework.Figure framework.ConnectionFigure.endFigure()	5

Continua na próxima página ...



... continuação da página anterior	
[Id] Método(Parâmetros)	fan-in
1466]framework.Figure framework.Connector.owner()	18
1467]Rectangle framework.Connector.displayBox()	9
1472]framework.FigureEnumeration framework.Drawing.figures()	9
1474]framework.FigureEnumeration framework.Drawing.figuresReverse()	5
1475]framework.Figure framework.Drawing.findFigure(int,int)	9
1479]framework.Figure framework.Drawing.findFigureInside(int,int)	6
1486]framework.Figure framework.Drawing.add(framework.Figure)	7
1489]framework.Figure framework.Drawing.remove(framework.Figure)	6
1490]framework.Figure framework.Drawing.orphan(framework.Figure)	7
1514]framework.DrawingView framework.DrawingEditor.view()	26
1516]framework.Tool framework.DrawingEditor.tool()	6
1517]void framework.DrawingEditor.toolDone()	20
1518]void framework.DrawingEditor.figureSelectionChanged(framework.DrawingView)	5
1521]void framework.DrawingEditor.showStatus(String)	6
1524]framework.Tool framework.DrawingView.tool()	5
1525]framework.Drawing framework.DrawingView.drawing()	62
1527]framework.DrawingEditor framework.DrawingView.editor()	6
1528]framework.Figure framework.DrawingView.add(framework.Figure)	10
1536]framework.FigureEnumeration framework.DrawingView.selection()	17
1538]int framework.DrawingView.selectionCount()	16
1539]boolean framework.DrawingView.isFigureSelected(framework.Figure)	7
1540]void framework.DrawingView.addToSelection(framework.Figure)	11
1542]void framework.DrawingView.addToSelectionAll(framework.FigureEnumeration)	7
1545]void framework.DrawingView.clearSelection()	31
1551]void framework.DrawingView.checkDamage()	28
1558]void framework.DrawingView.drawAll(Graphics)	6
1565]void framework.DrawingView.unfreezeView()	5
1570]boolean framework.DrawingView.isInteractive()	5
1571]void framework.Figure.moveBy(int,int)	13
1572]void framework.Figure.basicDisplayBox(Point,Point)	12
1573]void framework.Figure.displayBox(Point,Point)	14
1574]Rectangle framework.Figure.displayBox()	86
1575]void framework.Figure.draw(Graphics)	8
1576]framework.HandleEnumeration framework.Figure.handles()	6
1577]Dimension framework.Figure.size()	10
1578]Point framework.Figure.center()	9
1580]framework.FigureEnumeration framework.Figure.figures()	24
1582]boolean framework.Figure.containsPoint(int,int)	16
1583]Object framework.Figure.clone()	8
1585]boolean framework.Figure.includes(framework.Figure)	14
1587]void framework.Figure.addToContainer(framework.FigureChangeListener)	5
1588]void framework.Figure.removeFromContainer(framework.FigureChangeListener)	5
1590]void framework.Figure.removeDependendFigure(framework.Figure)	5
1592]framework.FigureChangeListener framework.Figure.listener()	21
1593]void framework.Figure.addFigureChangeListener(framework.FigureChangeListener)	11
1594]void framework.Figure.removeFigureChangeListener(framework.FigureChangeListener)	10
1595]void framework.Figure.release()	6
1596]void framework.Figure.invalidate()	9
1597]void framework.Figure.willChange()	25
1598]void framework.Figure.changed()	37
1599]boolean framework.Figure.canConnect()	9
1600]framework.Connector framework.Figure.connectorAt(int,int)	6
1601]void framework.Figure.connectorVisibility(boolean,framework.ConnectionFigure)	7
1602]Insets framework.Figure.connectionInsets()	5
1630]void framework.FigureChangeListener.figureInvalidated(framework.FigureChangeEvent)	5
1632]void framework.FigureChangeListener.figureRemoved(framework.FigureChangeEvent)	5
1633]void framework.FigureChangeListener.figureRequestRemove(framework.FigureChangeEvent)	6
1634]void framework.FigureChangeListener.figureRequestUpdate(framework.FigureChangeEvent)	6
1635]framework.Figure framework.FigureEnumeration.nextFigure()	95
1636]boolean framework.FigureEnumeration.hasNextFigure()	99
1640]void framework.FigureSelectionListener.figureSelectionChanged(framework.DrawingView)	5
1651]framework.Figure framework.Handle.owner()	37
1652]Rectangle framework.Handle.displayBox()	12
1657]framework.Handle framework.HandleEnumeration.nextHandle()	6
1658]boolean framework.HandleEnumeration.hasNextHandle()	7
1669]Point framework.Locator.locate(framework.Figure)	6
1675]void framework.Tool.activate()	12
1676]void framework.Tool.deactivate()	15
1677]void framework.Tool.mouseDown(event.MouseEvent,int,int)	22
1678]void framework.Tool.mouseDrag(event.MouseEvent,int,int)	11
1679]void framework.Tool.mouseUp(event.MouseEvent,int,int)	11
1684]boolean framework.Tool.isUsable()	5
1686]framework.DrawingEditor framework.Tool.editor()	30
1706]void samples.javadraw.AnimationDecorator.basicDisplayBox(Point,Point)	6
1707]Rectangle samples.javadraw.AnimationDecorator.displayBox()	50
1708]void samples.javadraw.AnimationDecorator.write(util.StorableOutput)	6
1709]void samples.javadraw.AnimationDecorator.read(util.StorableInput)	6
1715]framework.Figure samples.javadraw.BouncingDrawing.add(framework.Figure)	13
1716]framework.Figure samples.javadraw.BouncingDrawing.remove(framework.Figure)	8
1721]void samples.javadraw.FollowURLTool.mouseUp(event.MouseEvent,int,int)	9
1727]void samples.javadraw.JavaDrawApp.createTools(JToolBar)	6
1733]framework.Drawing samples.javadraw.JavaDrawApp.createDrawing()	5
1752]framework.DrawingView samples.javadraw.JavaDrawViewer.view()	6
1757]void samples.javadraw.JavaDrawViewer.toolDone()	14
1758]void samples.javadraw.JavaDrawViewer.figureSelectionChanged(framework.DrawingView)	5
1762]void samples.javadraw.MySelectionTool.mouseDown(event.MouseEvent,int,int)	17
1768]void samples.javadraw.URLTool.mouseDown(event.MouseEvent,int,int)	15
1769]void samples.javadraw.URLTool.mouseUp(event.MouseEvent,int,int)	9
1783]void samples.net.NetApp.createTools(JToolBar)	5
1786]Rectangle samples.net.NodeFigure.displayBox()	54
1787]boolean samples.net.NodeFigure.containsPoint(int,int)	16

Continua na próxima página ...

... continuação da página anterior		
[Id] Método(Parâmetros)	fan-in	
1789	void samples.net.NodeFigure.draw(Graphics)	5
1792	void samples.net.NodeFigure.connectorVisibility(boolean,framework.ConnectionFigure)	7
1793	framework.Connector samples.net.NodeFigure.connectorAt(int,int)	6
1800	void samples.nothing.NothingApp.createTools(JToolBar)	5
1807	void samples.pert.PertApplication.createTools(JToolBar)	5
1813	framework.HandleEnumeration samples.pert.PertDependency.handles()	5
1827	Rectangle samples.pert.PertFigure.displayBox()	48
1828	void samples.pert.PertFigure.basicDisplayBox(Point,Point)	5
1830	void samples.pert.PertFigure.draw(Graphics)	5
1837	void samples.pert.PertFigure.figureRemoved(framework.FigureChangeEvent)	5
1841	void samples.pert.PertFigure.write(util.StorableOutput)	7
1843	void samples.pert.PertFigure.read(util.StorableInput)	7
1856	framework.DrawingView standard.AbstractCommand.view()	42
1857	String standard.AbstractCommand.name()	7
1860	void standard.AbstractCommand.execute()	24
1861	boolean standard.AbstractCommand.isExecutable()	5
1879	framework.Figure standard.AbstractConnector.owner()	18
1883	Rectangle standard.AbstractConnector.displayBox()	9
1890	void standard.AbstractFigure.moveBy(int,int)	13
1891	void standard.AbstractFigure.basicMoveBy(int,int)	8
1892	void standard.AbstractFigure.displayBox(Point,Point)	14
1893	void standard.AbstractFigure.basicDisplayBox(Point,Point)	12
1894	Rectangle standard.AbstractFigure.displayBox()	86
1895	framework.HandleEnumeration standard.AbstractFigure.handles()	6
1896	framework.FigureEnumeration standard.AbstractFigure.figures()	24
1897	Dimension standard.AbstractFigure.size()	10
1900	boolean standard.AbstractFigure.containsPoint(int,int)	16
1902	boolean standard.AbstractFigure.includes(framework.Figure)	14
1904	void standard.AbstractFigure.addToContainer(framework.FigureChangeListener)	5
1905	void standard.AbstractFigure.removeFromContainer(framework.FigureChangeListener)	5
1906	void standard.AbstractFigure.addFigureChangeListener(framework.FigureChangeListener)	11
1907	void standard.AbstractFigure.removeFigureChangeListener(framework.FigureChangeListener)	10
1908	framework.FigureChangeListener standard.AbstractFigure.listener()	21
1909	void standard.AbstractFigure.release()	6
1910	void standard.AbstractFigure.invalidate()	9
1911	Rectangle standard.AbstractFigure.invalidateRectangle(Rectangle)	6
1912	void standard.AbstractFigure.willChange()	25
1913	void standard.AbstractFigure.changed()	37
1914	Point standard.AbstractFigure.center()	9
1915	boolean standard.AbstractFigure.canConnect()	9
1916	Insets standard.AbstractFigure.connectionInsets()	5
1917	framework.Connector standard.AbstractFigure.connectorAt(int,int)	6
1918	void standard.AbstractFigure.connectorVisibility(boolean,framework.ConnectionFigure)	7
1924	Object standard.AbstractFigure.clone()	8
1925	void standard.AbstractFigure.write(util.StorableOutput)	18
1926	void standard.AbstractFigure.read(util.StorableInput)	18
1932	void standard.AbstractFigure.removeDependentFigure(framework.Figure)	5
1942	framework.Figure standard.AbstractHandle.owner()	37
1943	Rectangle standard.AbstractHandle.displayBox()	12
1953	void standard.AbstractTool.activate()	12
1954	void standard.AbstractTool.deactivate()	15
1958	void standard.AbstractTool.mouseDown(event.MouseEvent,int,int)	22
1959	void standard.AbstractTool.mouseDrag(event.MouseEvent,int,int)	11
1960	void standard.AbstractTool.mouseUp(event.MouseEvent,int,int)	11
1963	framework.Drawing standard.AbstractTool.drawing()	19
1965	framework.DrawingEditor standard.AbstractTool.editor()	26
1967	framework.DrawingView standard.AbstractTool.view()	37
1970	boolean standard.AbstractTool.isUsable()	5
1997	void standard.ActionTool.mouseDown(event.MouseEvent,int,int)	16
1998	void standard.ActionTool.mouseUp(event.MouseEvent,int,int)	9
2007	void standard.AlignCommand.execute()	24
2012	boolean standard.AlignCommand\$UndoActivity.undo()	24
2013	boolean standard.AlignCommand\$UndoActivity.redo()	7
2022	void standard.BoxHandleKit.addHandles(framework.Figure,List)	7
2036	boolean standard.ResizeHandle\$UndoActivity.undo()	24
2037	boolean standard.ResizeHandle\$UndoActivity.redo()	7
2058	void standard.BringToFrontCommand.execute()	24
2066	void standard.ChangeAttributeCommand.execute()	24
2070	boolean standard.ChangeAttributeCommand\$UndoActivity.undo()	24
2071	boolean standard.ChangeAttributeCommand\$UndoActivity.redo()	7
2108	boolean standard.ChangeConnectionHandle\$UndoActivity.undo()	24
2109	boolean standard.ChangeConnectionHandle\$UndoActivity.redo()	7
2129	framework.Figure standard.CompositeFigure.add(framework.Figure)	7
2147	void standard.CompositeFigure.draw(Graphics)	5
2149	framework.Figure standard.CompositeFigure.figureAt(int)	5
2150	framework.FigureEnumeration standard.CompositeFigure.figures()	24
2153	boolean standard.CompositeFigure.containsFigure(framework.Figure)	5
2154	framework.FigureEnumeration standard.CompositeFigure.figuresReverse()	6
2161	boolean standard.CompositeFigure.includes(framework.Figure)	14
2163	void standard.CompositeFigure.release()	6
2165	void standard.CompositeFigure.figureRequestRemove(framework.FigureChangeEvent)	6
2166	void standard.CompositeFigure.figureRequestUpdate(framework.FigureChangeEvent)	6
2168	void standard.CompositeFigure.figureRemoved(framework.FigureChangeEvent)	5
2169	void standard.CompositeFigure.write(util.StorableOutput)	7
2170	void standard.CompositeFigure.read(util.StorableInput)	7
2192	void standard.ConnectionTool.mouseDown(event.MouseEvent,int,int)	15
2193	void standard.ConnectionTool.mouseDrag(event.MouseEvent,int,int)	9
2194	void standard.ConnectionTool.mouseUp(event.MouseEvent,int,int)	9
2195	void standard.ConnectionTool.deactivate()	15
2218	boolean standard.ConnectionTool\$UndoActivity.undo()	24
2219	boolean standard.ConnectionTool\$UndoActivity.redo()	7

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
2223	void standard.CopyCommand.execute()	24
2227	void standard.CreationTool.activate()	12
2228	void standard.CreationTool.mouseDown(event.MouseEvent,int,int)	19
2230	void standard.CreationTool.mouseDrag(event.MouseEvent,int,int)	10
2231	void standard.CreationTool.mouseUp(event.MouseEvent,int,int)	9
2240	void standard.CutCommand.execute()	24
2244	boolean standard.CutCommand\$UndoActivity.undo()	24
2245	boolean standard.CutCommand\$UndoActivity.redo()	7
2249	Insets standard.DecoratorFigure.connectionInsets()	5
2250	boolean standard.DecoratorFigure.canConnect()	9
2251	boolean standard.DecoratorFigure.containsPoint(int,int)	16
2256	Rectangle standard.DecoratorFigure.displayBox()	51
2257	void standard.DecoratorFigure.basicDisplayBox(Point,Point)	6
2261	boolean standard.DecoratorFigure.includes(figure.framework.Figure)	13
2262	void standard.DecoratorFigure.moveBy(int,int)	13
2264	void standard.DecoratorFigure.release()	6
2265	void standard.DecoratorFigure.figureInvalidated(figure.framework.FigureChangeEvent)	5
2267	void standard.DecoratorFigure.figureRemoved(figure.framework.FigureChangeEvent)	5
2268	void standard.DecoratorFigure.figureRequestUpdate(figure.framework.FigureChangeEvent)	6
2269	void standard.DecoratorFigure.figureRequestRemove(figure.framework.FigureChangeEvent)	6
2270	framework.FigureEnumeration standard.DecoratorFigure.figures()	9
2277	framework.Connector standard.DecoratorFigure.connectorAt(int,int)	6
2278	void standard.DecoratorFigure.connectorVisibility(boolean,framework.ConnectionFigure)	7
2279	void standard.DecoratorFigure.write(util.StorableOutput)	6
2280	void standard.DecoratorFigure.read(util.StorableInput)	6
2285	void standard.DecoratorFigure.removeDependentFigure(figure.framework.Figure)	5
2287	void standard.DeleteCommand.execute()	24
2291	boolean standard.DeleteCommand\$UndoActivity.undo()	25
2292	boolean standard.DeleteCommand\$UndoActivity.redo()	8
2301	void standard.DragTracker.mouseDown(event.MouseEvent,int,int)	15
2302	void standard.DragTracker.mouseDrag(event.MouseEvent,int,int)	9
2303	void standard.DragTracker.activate()	9
2304	void standard.DragTracker.deactivate()	14
2307	boolean standard.DragTracker\$UndoActivity.undo()	24
2308	boolean standard.DragTracker\$UndoActivity.redo()	7
2315	void standard.DuplicateCommand.execute()	24
2322	framework.Figure standard.FigureAndEnumerator.nextFigure()	95
2323	boolean standard.FigureAndEnumerator.hasNextFigure()	99
2328	void standard.FigureChangeAdapter.figureRemoved(figure.framework.FigureChangeEvent)	5
2329	void standard.FigureChangeAdapter.figureRequestRemove(figure.framework.FigureChangeEvent)	6
2330	void standard.FigureChangeAdapter.figureRequestUpdate(figure.framework.FigureChangeEvent)	6
2333	void standard.FigureChangeEventMulticaster.figureRequestRemove(figure.framework.FigureChangeEvent)	6
2334	void standard.FigureChangeEventMulticaster.figureRequestUpdate(figure.framework.FigureChangeEvent)	6
2336	void standard.FigureChangeEventMulticaster.figureRemoved(figure.framework.FigureChangeEvent)	5
2343	boolean standard.FigureEnumerator.hasNextFigure()	99
2344	framework.Figure standard.FigureEnumerator.nextFigure()	95
2356	framework.Handle standard.HandleAndEnumerator.nextHandle()	5
2357	boolean standard.HandleAndEnumerator.hasNextHandle()	6
2361	boolean standard.HandleEnumerator.hasNextHandle()	7
2362	framework.Handle standard.HandleEnumerator.nextHandle()	6
2367	void standard.HandleTracker.mouseDown(event.MouseEvent,int,int)	15
2368	void standard.HandleTracker.mouseDrag(event.MouseEvent,int,int)	9
2369	void standard.HandleTracker.mouseUp(event.MouseEvent,int,int)	9
2370	void standard.HandleTracker.activate()	9
2381	Rectangle standard.LocatorConnector.displayBox()	9
2393	framework.Drawing standard.NullDrawingView.drawing()	45
2396	framework.Figure standard.NullDrawingView.add(figure.framework.Figure)	9
2404	framework.FigureEnumeration standard.NullDrawingView.selection()	14
2406	int standard.NullDrawingView.selectionCount()	13
2408	void standard.NullDrawingView.addToSelection(figure.framework.Figure)	9
2410	void standard.NullDrawingView.addToSelectionAll(figure.framework.FigureEnumeration)	5
2413	void standard.NullDrawingView.clearSelection()	30
2419	void standard.NullDrawingView.checkDamage()	23
2426	void standard.NullDrawingView.drawAll(Graphics)	6
2433	void standard.NullDrawingView.unfreezeView()	5
2441	boolean standard.NullDrawingView.isInteractive()	5
2446	void standard.NullTool.activate()	9
2447	void standard.NullTool.deactivate()	14
2452	Point standard.OffsetLocator.locate(figure.framework.Figure)	6
2461	void standard.PasteCommand.execute()	24
2466	boolean standard.PasteCommand\$UndoActivity.undo()	24
2467	boolean standard.PasteCommand\$UndoActivity.redo()	7
2493	framework.Locator standard.RelativeLocator.northEast()	6
2494	framework.Locator standard.RelativeLocator.northWest()	6
2496	framework.Locator standard.RelativeLocator.southEast()	6
2497	framework.Locator standard.RelativeLocator.southWest()	6
2500	boolean standard.ReverseFigureEnumerator.hasNextFigure()	99
2501	framework.Figure standard.ReverseFigureEnumerator.nextFigure()	95
2504	void standard.SelectAllCommand.execute()	24
2508	boolean standard.SelectAllCommand\$UndoActivity.undo()	24
2509	boolean standard.SelectAllCommand\$UndoActivity.redo()	7
2512	void standard.SelectAreaTracker.mouseDown(event.MouseEvent,int,int)	15
2513	void standard.SelectAreaTracker.mouseDrag(event.MouseEvent,int,int)	9
2514	void standard.SelectAreaTracker.mouseUp(event.MouseEvent,int,int)	9
2520	void standard.SelectionTool.mouseDown(event.MouseEvent,int,int)	17
2522	void standard.SelectionTool.mouseDrag(event.MouseEvent,int,int)	10
2523	void standard.SelectionTool.mouseUp(event.MouseEvent,int,int)	10
2530	void standard.SendToBackCommand.execute()	24
2534	boolean standard.SendToBackCommand\$UndoActivity.undo()	24
2535	boolean standard.SendToBackCommand\$UndoActivity.redo()	7
2543	boolean standard.SingleFigureEnumerator.hasNextFigure()	99

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
2544	framework.Figure standard.SingleFigureEnumerator.nextFigure()	95
2550	framework.Figure standard.StandardDrawing.orphan(framework.Figure)	10
2551	framework.Figure standard.StandardDrawing.add(framework.Figure)	13
2554	void standard.StandardDrawing.figureRequestUpdate(framework.FigureChangeEvent)	6
2556	Rectangle standard.StandardDrawing.displayBox()	52
2557	void standard.StandardDrawing.basicDisplayBox(Point,Point)	5
2570	framework.Tool standard.StandardDrawingView.tool()	5
2571	framework.Drawing standard.StandardDrawingView.drawing()	62
2573	framework.DrawingEditor standard.StandardDrawingView.editor()	5
2574	framework.Figure standard.StandardDrawingView.add(framework.Figure)	10
2584	framework.FigureEnumeration standard.StandardDrawingView.selection()	16
2586	int standard.StandardDrawingView.selectionCount()	16
2587	boolean standard.StandardDrawingView.isFigureSelected(framework.Figure)	7
2588	void standard.StandardDrawingView.addToSelection(framework.Figure)	11
2590	void standard.StandardDrawingView.addToSelectionAll(framework.FigureEnumeration)	7
2593	void standard.StandardDrawingView.clearSelection()	31
2606	void standard.StandardDrawingView.checkDamage()	28
2612	void standard.StandardDrawingView.drawAll(Graphics)	6
2625	void standard.StandardDrawingView.unfreezeView()	5
2630	boolean standard.StandardDrawingView.isInteractive()	5
2674	void standard.ToggleGridCommand.execute()	24
2676	framework.Tool standard.ToolButton.tool()	6
2677	String standard.ToolButton.name()	6
2747	List util.collections.jdk11.CollectionsFactoryJDK11.createList()	41
2749	List util.collections.jdk11.CollectionsFactoryJDK11.createList(int)	26
2814	List util.collections.jdk12.CollectionsFactoryJDK12.createList()	41
2816	List util.collections.jdk12.CollectionsFactoryJDK12.createList(int)	26
2822	List util.CollectionsFactory.createList()	41
2824	List util.CollectionsFactory.createList(int)	26
2829	util.CollectionsFactory util.CollectionsFactory.current()	76
2841	void util.Command.execute()	24
2842	boolean util.Command.isExecutable()	5
2843	String util.Command.name()	7
2858	void util.CommandMenu.add(util.Command)	14
2888	Point util.Geom.center(Rectangle)	7
2897	int util.Geom.range(int,int,int)	6
2898	long util.Geom.length2(int,int,int,int)	5
2899	long util.Geom.length(int,int,int,int)	5
2918	util.Iconkit util.Iconkit.instance()	6
2949	String util.PaletteButton.name()	6
2976	void util.RedoCommand.execute()	24
3008	void util.Storable.write(util.StorableOutput)	28
3009	void util.Storable.read(util.StorableInput)	28
3011	util.Storable util.StorableInput.readStorable()	20
3012	String util.StorableInput.readString()	10
3013	int util.StorableInput.readInt()	22
3017	boolean util.StorableInput.readBoolean()	5
3022	void util.StorableOutput.writeStorable(util.Storable)	18
3023	void util.StorableOutput.writeInt(int)	21
3027	void util.StorableOutput.writeBoolean(boolean)	5
3028	void util.StorableOutput.writeString(String)	10
3036	void util.StorableOutput.space()	8
3048	boolean util.Undoable.undo()	26
3049	boolean util.Undoable.redo()	9
3050	boolean util.Undoable.isUndoable()	9
3052	boolean util.Undoable.isRedoable()	24
3060	boolean util.UndoableAdapter.undo()	26
3061	boolean util.UndoableAdapter.redo()	9
3062	boolean util.UndoableAdapter.isUndoable()	8
3064	boolean util.UndoableAdapter.isRedoable()	23
3075	void util.UndoableCommand.execute()	7
3076	boolean util.UndoableCommand.isExecutable()	5
3079	String util.UndoableCommand.name()	7
3112	void util.UndoableTool.activate()	5
3113	void util.UndoableTool.deactivate()	5
3114	void util.UndoableTool.mouseDown(event.MouseEvent,int,int)	5
3115	void util.UndoableTool.mouseDrag(event.MouseEvent,int,int)	5
3116	void util.UndoableTool.mouseUp(event.MouseEvent,int,int)	5
3126	framework.DrawingEditor util.UndoableTool.editor()	5
3143	void util.UndoCommand.execute()	24
3167	boolean util.UndoRedoActivity.isUndoable()	8
3169	boolean util.UndoRedoActivity.isRedoable()	5

Na Tabela C.2 são apresentados os grupos obtidos pela instância 12 da fase de agrupamento no sistema JHotDraw com o respectivo resultado do cálculo de pontuação de cada uma das medidas de ordenação. A Coluna IT? indica se o grupo contém métodos referentes a instâncias de interesses transversais validadas, conforme Tabela A.1. A Coluna Métodos apresenta os identificadores dos métodos que compõem cada grupo, sendo que os métodos podem ser consultados na Tabela C.1.

Tabela C.2: Lista de grupos da instância 12 com pontuação: JHotDraw

Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K1		14,0000	14,0000	7,0000	7,0000	[53]
K2		4,0000	4,0000	2,0000	2,0000	[57]
K3		5,0000	5,0000	3,0000	3,0000	[77]
K4		4,0000	4,0000	3,0000	3,0000	[109]
K5		6,0000	6,0000	5,0000	5,0000	[186]
K6		3,0000	3,0000	2,0000	2,0000	[208]
K7		3,0000	3,0000	2,0000	2,0000	[210]
K8		4,0000	4,0000	3,0000	3,0000	[244]
K9		4,0000	4,0000	3,0000	3,0000	[268]
K10		3,0000	3,0000	2,0000	2,0000	[269]
K11		4,0000	4,0000	3,0000	3,0000	[282]
K12		4,0000	4,0000	2,0000	2,0000	[306]
K13		3,0000	3,0000	2,0000	2,0000	[313]
K14		3,0000	3,0000	2,0000	2,0000	[356]
K15		3,0000	3,0000	2,0000	2,0000	[406]
K16		3,0000	3,0000	2,0000	2,0000	[443]
K17		4,0000	4,0000	3,0000	3,0000	[463]
K18		5,0000	5,0000	4,0000	4,0000	[465]
K19		6,0000	6,0000	5,0000	5,0000	[469]
K20		3,0000	3,0000	2,0000	2,0000	[474]
K21		8,0000	8,0000	7,0000	7,0000	[607]
K22		5,0000	5,0000	3,0000	3,0000	[616]
K23		4,0000	4,0000	2,0000	2,0000	[642]
K24		6,0000	6,0000	4,0000	4,0000	[694]
K25		6,0000	6,0000	3,0000	3,0000	[695]
K26		3,0000	3,0000	2,0000	2,0000	[722]
K27		4,0000	4,0000	3,0000	3,0000	[749]
K28		3,0000	3,0000	2,0000	2,0000	[789]
K29		5,0000	5,0000	3,0000	3,0000	[806]
K30		3,0000	3,0000	2,0000	2,0000	[853]
K31		3,0000	3,0000	2,0000	2,0000	[913]
K32		3,0000	3,0000	2,0000	2,0000	[987]
K33		3,0000	3,0000	2,0000	2,0000	[991]
K34		3,0000	3,0000	2,0000	2,0000	[1038]
K35		5,0000	5,0000	3,0000	3,0000	[1091]
K36		4,0000	4,0000	2,0000	2,0000	[1125]
K37		3,0000	3,0000	2,0000	2,0000	[1136]
K38		4,0000	4,0000	2,0000	2,0000	[1180]
K39		4,0000	4,0000	3,0000	3,0000	[1263]
K40		9,0000	9,0000	5,0000	5,0000	[1264]
K41		22,0000	22,0000	15,0000	15,0000	[1270]
K42		8,0000	8,0000	4,0000	4,0000	[1288]
K43		3,0000	3,0000	2,0000	2,0000	[1318]
K44		3,0000	3,0000	2,0000	2,0000	[1348]
K45		3,0000	3,0000	2,0000	2,0000	[1359]
K46		5,0000	5,0000	3,0000	3,0000	[1362]
K47		5,0000	5,0000	4,0000	4,0000	[1382]
K48		4,0000	4,0000	2,0000	2,0000	[1415]
K49		4,0000	4,0000	2,0000	2,0000	[1430]
K50		15,0000	15,0000	9,0000	9,0000	[1472]
K51		10,0000	10,0000	5,0000	5,0000	[1474]
K52		18,0000	18,0000	9,0000	9,0000	[1475]
K53		12,0000	12,0000	6,0000	6,0000	[1479]
K54		5,0000	5,0000	2,0000	2,0000	[1481]
K55		5,0000	5,0000	2,0000	2,0000	[1496]
K56		5,0000	5,0000	2,0000	2,0000	[1497]
K57		5,0000	5,0000	2,0000	2,0000	[1501]
K58		8,0000	8,0000	5,0000	5,0000	[1587]
K59		8,0000	8,0000	5,0000	5,0000	[1588]
K60		7,0000	7,0000	4,0000	4,0000	[1619]
K61		5,0000	5,0000	2,0000	2,0000	[1653]
K62		5,0000	5,0000	2,0000	2,0000	[1654]
K63		5,0000	5,0000	2,0000	2,0000	[1711]
K64		5,0000	5,0000	2,0000	2,0000	[1712]
K65		4,0000	4,0000	2,0000	2,0000	[1717]
K66		4,0000	4,0000	3,0000	3,0000	[1772]
K67		3,0000	3,0000	2,0000	2,0000	[1816]
K68		3,0000	3,0000	2,0000	2,0000	[1823]
K69		4,0000	4,0000	3,0000	3,0000	[1833]
K70		3,0000	3,0000	2,0000	2,0000	[1835]
K71		5,0000	5,0000	3,0000	3,0000	[1839]
K72		3,0000	3,0000	2,0000	2,0000	[1862]
K73		3,0000	3,0000	2,0000	2,0000	[1882]
K74		37,0000	37,0000	19,0000	19,0000	[1963]
K75		56,0000	56,0000	37,0000	37,0000	[1967]
K76		15,0000	15,0000	7,0000	7,0000	[2022]
K77		3,0000	3,0000	2,0000	2,0000	[2038]
K78		3,0000	3,0000	2,0000	2,0000	[2110]
K79		13,0000	13,0000	7,0000	7,0000	[2129]
K80		7,0000	7,0000	5,0000	5,0000	[2149]
K81		6,0000	6,0000	5,0000	5,0000	[2153]
K82		4,0000	4,0000	3,0000	3,0000	[2173]
K83		3,0000	3,0000	2,0000	2,0000	[2202]
K84		7,0000	7,0000	3,0000	3,0000	[2349]
K85		5,0000	5,0000	2,0000	2,0000	[2453]
K86		5,0000	5,0000	4,0000	4,0000	[2597]
K87		9,0000	9,0000	6,0000	6,0000	[2676]
K88		5,0000	5,0000	3,0000	3,0000	[2713]
K89		3,0000	3,0000	2,0000	2,0000	[2718]
K90		4,0000	4,0000	2,0000	2,0000	[2837]

Continua na próxima página ...

... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIrank	Métodos
K91		5,0000	5,0000	2,0000	2,0000	[2840]
K92		5,0000	5,0000	2,0000	2,0000	[2865]
K93		4,0000	4,0000	2,0000	2,0000	[2877]
K94		5,0000	5,0000	2,0000	2,0000	[2883]
K95		4,0000	4,0000	2,0000	2,0000	[2885]
K96		14,0000	14,0000	7,0000	7,0000	[2888]
K97		4,0000	4,0000	2,0000	2,0000	[2892]
K98		11,0000	11,0000	6,0000	6,0000	[2897]
K99		10,0000	10,0000	5,0000	5,0000	[2898]
K100		10,0000	10,0000	5,0000	5,0000	[2899]
K101		5,0000	5,0000	2,0000	2,0000	[2904]
K102		4,0000	4,0000	2,0000	2,0000	[2916]
K103		11,0000	11,0000	6,0000	6,0000	[2918]
K104		5,0000	5,0000	3,0000	3,0000	[2920]
K105		5,0000	5,0000	2,0000	2,0000	[2921]
K106		4,0000	4,0000	2,0000	2,0000	[2924]
K107		41,0000	41,0000	20,0000	20,0000	[3011]
K108		20,0000	20,0000	10,0000	10,0000	[3012]
K109		41,0000	41,0000	22,0000	22,0000	[3013]
K110		5,0000	5,0000	2,0000	2,0000	[3016]
K111		11,0000	11,0000	5,0000	5,0000	[3017]
K112		37,0000	37,0000	18,0000	18,0000	[3022]
K113		5,0000	5,0000	2,0000	2,0000	[3029]
K114		9,0000	9,0000	8,0000	8,0000	[3036]
K115		5,0000	5,0000	2,0000	2,0000	[3041]
K116		3,0000	3,0000	2,0000	2,0000	[3154]
K117		3,0000	3,0000	2,0000	2,0000	[3161]
K118		3,0000	3,0000	2,0000	2,0000	[3186]
K119		7,0000	4,7833	4,0000	2,7333	[1200] [1201]
K120		4,0000	2,7200	4,0000	2,7200	[896] [897]
K121		4,0000	2,7200	4,0000	2,7200	[2515] [2516]
K122		6,0000	4,0091	4,0000	2,6727	[599] [601]
K123		7,0000	4,6262	5,0000	3,3044	[363] [364]
K124		13,0000	8,3400	7,0000	4,4908	[1836] [1837]
K125		4,0000	2,5412	4,0000	2,5412	[704] [1301]
K126		9,0000	5,4318	12,0000	7,2424	[51] [622]
K127		6,0000	3,5229	4,0000	2,3486	[1735] [1744]
K128		6,0000	3,4729	4,0000	2,3153	[898] [2517]
K129		6,0000	3,4375	4,0000	2,2916	[2174] [2474]
K130		5,0000	2,8412	6,0000	3,4094	[1073] [1414]
K131		16,0000	8,9937	14,0000	7,8695	[673] [686]
K132		42,0000	23,4652	30,0000	16,7609	[2150] [2154]
K133		4,0000	2,2119	4,0000	2,2119	[3046] [3047]
K134		44,0000	24,4147	78,0000	43,2806	[2824] [2749] [2816]
K135		7,0000	3,8422	4,0000	2,1956	[1186] [1187]
K136	Sim	47,0000	25,7616	47,0000	25,7616	[3052] [3064]
K137	Sim	55,0000	30,2447	92,0000	50,5911	[2413] [1545] [2593]
K138		59,0000	31,8296	74,0000	39,9219	[1651] [1942]
K139		4,0000	2,1547	4,0000	2,1547	[56] [1729]
K140		6,0000	3,2143	4,0000	2,1429	[291] [292]
K141		30,0000	16,0279	36,0000	19,2335	[1466] [1879]
K142		54,0000	31,3955	38,0000	22,0931	[3026] [3027] [3023] [3028]
K143		5,0000	2,6706	4,0000	2,1365	[99] [100]
K144		7,0000	3,7333	4,0000	2,1333	[1939] [1940]
K145		8,0000	4,2600	5,0000	2,6625	[1207] [1208]
K146		11,0000	5,8562	12,0000	6,3886	[1291] [1911]
K147	Sim	54,0000	35,6200	632,0000	416,8860	[1178] [1428] [1065] [2508] [2307] [2108] [714] [882] [1134] [1299] [1316] [2036] [1080] [1085] [2012] [2070] [2244] [2291] [1147] [1441] [2466] [2534] [851] [2218] [3048] [3060]
K148		30,0000	15,8464	42,0000	22,1850	[1592] [1908]
K149		7,0000	3,6958	4,0000	2,1119	[989] [997]
K150		28,0000	14,7200	28,0000	14,7200	[1573] [1892]
K151		6,0000	3,1379	4,0000	2,0920	[1873] [1874]
K152		24,0000	12,4695	24,0000	12,4695	[1652] [1943]
K153		6,0000	3,1031	4,0000	2,0687	[1204] [1205]
K154		32,0000	16,5092	33,0000	17,0251	[1536] [2584]
K155		33,0000	17,9133	144,0000	78,1670	[1582] [677] [1279] [866] [257] [441] [2251] [1787] [1900]
K156		22,0000	12,0168	36,0000	19,6638	[1446] [1090] [1206]
K157		20,0000	10,2234	20,0000	10,2234	[1577] [1897]
K158		16,0000	11,5634	22,0000	15,8997	[1944] [879] [2100] [2185] [711] [1131] [1945] [1158] [1313] [1102] [2444]
K159		11,0000	5,6050	10,0000	5,0955	[3112] [3113]
K160		17,0000	9,3303	27,0000	14,8187	[1467] [1883] [2381]
K161		156,0000	85,5654	970,0000	532,0410	[1635] [2344] [2501] [2544] [2322] [2323] [1636] [2343] [2500] [2543]
K162		48,0000	24,1920	56,0000	28,2240	[1686] [1965]
K163		13,0000	6,7215	24,0000	12,4089	[2493] [2494] [2496] [2497]
K164		4,0000	1,9792	4,0000	1,9792	[2182] [2183]
K165		4,0000	1,9750	4,0000	1,9750	[753] [1780]
K166	Sim	56,0000	28,7786	79,0000	40,5984	[2419] [1551] [2606]
K167		7,0000	3,4460	4,0000	1,9691	[237] [238]
K168	Sim	22,0000	11,1197	29,0000	14,6578	[2396] [1528] [2574]
K169		94,0000	48,5439	169,0000	87,2757	[2393] [1525] [2571]
K170	Sim	17,0000	8,2924	17,0000	8,2924	[3050] [3062]
K171	Sim	29,0000	15,9852	75,0000	41,3410	[1571] [442] [2262] [1890] [805] [1361]
K172		18,0000	11,2441	48,0000	29,9843	[1583] [1924] [314] [438] [869] [1238]
K173		30,0000	15,0760	45,0000	22,6140	[2406] [1538] [2586]
K174		23,0000	11,5253	31,0000	15,5341	[2408] [1540] [2588]
K175	Sim	15,0000	7,3166	27,0000	13,1698	[1596] [1242] [1910]
K176		4,0000	1,9086	4,0000	1,9086	[2098] [2101]
K177		8,0000	3,8154	6,0000	2,8615	[701] [1105]
K178		10,0000	4,7474	10,0000	4,7474	[1524] [2570]

Continua na próxima página ...

... continuação da página anterior									
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos			
K179		15,0000	7,1832	18,0000	8,6198	[2612]	[1558]	[2426]	
K180		8,0000	3,7760	10,0000	4,7200	[80]	[1733]		
K181		4,0000	1,8880	4,0000	1,8880	[110]	[1726]		
K182		31,0000	17,6788	158,0000	90,1051	[2304]	[721]	[1347]	[294]
K183		4,0000	1,8800	4,0000	1,8800	[450]	[810]		
K184		24,0000	12,4954	68,0000	35,4037	[1239]	[1585]	[1902]	[2161]
K185		7,0000	3,9853	9,0000	5,1240	[1950]	[2454]	[2488]	
K186		7,0000	3,9853	9,0000	5,1240	[1951]	[2455]	[2489]	
K187		12,0000	5,6030	10,0000	4,6692	[1904]	[1905]		
K188		8,0000	3,7279	5,0000	2,3300	[3000]	[3001]		
K189		70,0000	37,2470	50,0000	26,6050	[1236]	[1896]	[2404]	[2437]
K190	Sim	12,0000	5,8970	15,0000	7,3713	[2842]	[1861]	[3076]	
K191		13,0000	6,3884	21,0000	10,3198	[2843]	[1857]	[3079]	
K192		10,0000	4,6000	10,0000	4,6000	[1684]	[1970]		
K193		14,0000	6,7826	28,0000	13,5651	[1918]	[2278]	[1601]	[1792]
K194		15,0000	7,9345	48,0000	25,3904	[1909]	[1213]	[1395]	[2163]
K195		7,0000	4,0376	6,0000	3,4608	[849]	[1078]	[1426]	
K196		12,0000	5,4696	15,0000	6,8370	[2630]	[1570]	[2441]	
K197		19,0000	8,9708	54,0000	25,4960	[1342]	[1917]	[1115]	[678]
K198		18,0000	8,7479	45,0000	21,8698	[1182]	[1915]	[2250]	[1151]
K199		19,0000	8,4790	17,0000	7,5864	[1490]	[2550]		
K200		17,0000	9,4299	60,0000	33,2819	[1391]	[488]	[331]	[2333]
K201		18,0000	9,0922	60,0000	30,3073	[1393]	[489]	[825]	[1634]
K202		8,0000	3,5520	12,0000	5,3280	[106]	[1521]		
K203		6,0000	2,6600	6,0000	2,6600	[2196]	[2203]		
K204		8,0000	3,5312	6,0000	2,6484	[1569]	[2578]		
K205		8,0000	3,6563	12,0000	5,4844	[1674]	[1412]	[1980]	
K206		5,0000	2,1786	4,0000	1,7429	[2950]	[2951]		
K207		6,0000	2,6100	8,0000	3,4800	[378]	[524]		
K208		17,0000	7,3893	14,0000	6,0853	[1489]	[1716]		
K209		10,0000	4,3436	12,0000	5,2123	[2677]	[2949]		
K210		34,0000	18,1530	197,0000	105,1805	[1679]	[1960]	[2523]	[298]
K211		30,0000	15,4794	162,0000	83,5889	[836]	[1408]	[1678]	[1959]
K212		15,0000	7,9177	45,0000	23,7531	[2267]	[487]	[2336]	[1390]
K213		15,0000	6,6364	25,0000	11,0606	[1518]	[25]	[98]	[1640]
K214		8,0000	3,7538	9,0000	4,2230	[1659]	[2358]	[2363]	
K215		10,0000	4,3238	10,0000	4,3238	[1590]	[1932]		
K216		16,0000	7,7872	40,0000	19,4681	[1961]	[746]	[2191]	[1680]
K217		8,0000	3,4301	7,0000	3,0013	[2198]	[2205]		
K218		14,0000	6,3901	19,0000	8,6723	[2410]	[1542]	[2590]	
K219		7,0000	2,9952	5,0000	2,1394	[179]	[660]		
K220		18,0000	9,5687	45,0000	23,9217	[2169]	[333]	[1841]	[1708]
K221		7,0000	2,9867	8,0000	3,4133	[1682]	[1973]		
K222		8,0000	3,7710	9,0000	4,2424	[3054]	[2078]	[3070]	
K223		10,0000	4,7858	12,0000	5,7430	[1610]	[1217]	[1929]	
K224		13,0000	5,8589	12,0000	5,4083	[1990]	[1991]	[1988]	[1989]
K225		6,0000	2,5484	6,0000	2,5484	[1537]	[2585]		
K226		6,0000	3,1184	9,0000	4,6776	[1552]	[995]	[2607]	
K227		18,0000	9,4367	45,0000	23,5917	[2170]	[332]	[1843]	[1709]
K228		5,0000	2,1076	4,0000	1,6861	[1660]	[2364]		
K229		9,0000	3,7823	8,0000	3,3621	[1589]	[1931]		
K230		10,0000	4,8633	20,0000	9,7267	[1899]	[2159]	[2259]	[1237]
K231		6,0000	2,9354	9,0000	4,4032	[2090]	[2081]	[2115]	
K232	Sim	38,0000	21,1510	190,0000	105,7548	[3049]	[3061]	[2013]	[1135]
K233		5,0000	2,0833	4,0000	1,6667	[1705]	[2263]		
K234		36,0000	17,3650	68,0000	32,8006	[1757]	[1517]	[24]	[97]
K235		5,0000	2,0778	5,0000	2,0778	[841]	[843]		
K236		9,0000	3,7699	9,0000	3,7699	[2575]	[1529]	[2397]	
K237		9,0000	3,7699	9,0000	3,7699	[2592]	[1544]	[2412]	
K238		20,0000	9,7320	34,0000	16,5444	[671]	[867]	[1578]	[1914]
K239		11,0000	4,5853	12,0000	5,0021	[1812]	[1197]	[1451]	
K240		8,0000	3,3013	9,0000	3,7139	[2823]	[2748]	[2815]	
K241		8,0000	3,3013	9,0000	3,7139	[2825]	[2750]	[2817]	
K242		8,0000	3,9396	9,0000	4,4320	[2847]	[1866]	[3085]	
K243		7,0000	2,8436	6,0000	2,4373	[1547]	[2415]		
K244		48,0000	26,2995	403,0000	220,8062	[175]	[655]	[2228]	[1072]
K245		9,0000	3,6409	7,0000	2,8318	[3147]	[3160]		
K246		5,0000	2,0187	4,0000	1,6150	[405]	[506]		
K247		5,0000	2,0165	4,0000	1,6132	[1562]	[2616]		
K248		12,0000	6,2328	34,0000	17,6595	[445]	[793]	[444]	[446]
K249		5,0000	2,0065	4,0000	1,6052	[1847]	[2229]		
K250		5,0000	2,0000	4,0000	1,6000	[1838]	[1840]		
K251		37,0000	14,7962	42,0000	16,7957	[94]	[1514]		
K252		18,0000	8,5235	51,0000	24,1499	[1645]	[1936]	[1937]	[706]
K253		7,0000	2,8077	6,0000	2,4066	[2600]	[1548]	[2416]	
K254		28,0000	14,1938	125,0000	63,3650	[1675]	[1076]	[839]	[1411]
K255		141,0000	72,5204	1084,0000	557,5326	[1574]	[1894]	[1707]	[1152]
K256	Sim	26,0000	12,1134	33,0000	15,3747	[1486]	[1715]	[2551]	
K257		5,0000	1,9714	4,0000	1,5771	[1647]	[1938]		
K258		5,0000	1,9714	4,0000	1,5771	[1650]	[1941]		
K259	Sim	54,0000	26,8124	161,0000	79,9406	[1598]	[1371]	[1913]	[1597]
K260		7,0000	2,7316	6,0000	2,3413	[1483]	[2547]		

Continua na próxima página ...

... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K261		6,0000	2,3396	4,0000	1,5598	[3149] [3175]
K262		6,0000	2,3276	4,0000	1,5517	[1584] [1901]
K263		12,0000	5,6646	37,0000	17,4659	[2356] [2357] [1658] [2361] [1657] [2362]
K264		7,0000	2,7330	6,0000	2,3426	[2827] [2752] [2819]
K265		4,0000	1,5467	4,0000	1,5467	[1458] [1459]
K266		6,0000	2,3100	4,0000	1,5400	[380] [518]
K267		6,0000	2,3100	6,0000	2,3100	[381] [519]
K268		6,0000	2,3100	4,0000	1,5400	[1671] [2352]
K269		22,0000	9,6259	18,0000	7,8757	[3167] [3169] [3165] [3166]
K270		9,0000	3,8306	12,0000	5,1074	[1093] [1603] [1919] [2276]
K271		6,0000	2,3040	4,0000	1,5360	[1554] [2422]
K272		6,0000	2,3040	4,0000	1,5360	[3003] [3179]
K273		8,0000	3,1332	9,0000	3,5249	[2667] [792] [1378]
K274		6,0000	2,5473	6,0000	2,5473	[1962] [1681] [3118]
K275		15,0000	8,3509	43,0000	23,9392	[419] [501] [537] [338] [343] [350] [545]
K276		15,0000	8,3509	43,0000	23,9392	[420] [502] [538] [339] [344] [351] [546]
K277		12,0000	5,7029	12,0000	5,7029	[3098] [3095] [3096] [3103] [3104]
K278		7,0000	2,6521	6,0000	2,2733	[2613] [1559] [2427]
K279		17,0000	7,7619	47,0000	21,4593	[1630] [2332] [2164] [2265] [1392] [485] [820] [1057] [2552] [1211] [2326]
K280		16,0000	6,9205	40,0000	17,3014	[1916] [1602] [1058] [2249] [1341] [256] [865] [1114] [1844]
K281		3,0000	1,1200	4,0000	1,4933	[2740] [2741]
K282		7,0000	2,6123	4,0000	1,4927	[2018] [2313]
K283		9,0000	6,0059	20,0000	13,3463	[15] [13] [1803] [1740] [1805]
K284		14,0000	5,6703	20,0000	8,1004	[93] [1516] [21] [1755]
K285		13,0000	5,6705	7,0000	3,0533	[2216] [1063] [2238]
K286		10,0000	4,2259	12,0000	5,0711	[1235] [1579] [829] [665] [1259] [1898]
K287		17,0000	7,7900	28,0000	12,8306	[1669] [2452] [1307] [1096] [2470] [2487]
K288		7,0000	2,6820	6,0000	2,2989	[1641] [2296] [2374]
K289		12,0000	4,3457	7,0000	2,5350	[683] [684]
K290		15,0000	5,4265	12,0000	4,3412	[1453] [1454]
K291		7,0000	2,5860	6,0000	2,2166	[2664] [788] [1365]
K292		7,0000	3,5250	8,0000	4,0286	[2835] [2836] [2838]
K293		6,0000	2,6743	6,0000	2,6743	[1689] [1981] [3136]
K294		4,0000	1,4400	4,0000	1,4400	[1986] [2448]
K295		8,0000	3,2133	13,0000	5,2217	[2490] [2492] [2491] [2495]
K296		15,0000	5,3741	8,0000	2,8662	[2348] [2350]
K297		5,0000	1,7821	4,0000	1,4257	[382] [520]
K298		12,0000	4,9470	18,0000	7,4205	[1649] [2033] [1295] [699] [708] [876] [3099] [2097] [2179]
K299		20,0000	9,8233	32,0000	15,7173	[1099] [1648] [2096] [2178] [875] [2042] [2048] [2050] [2054] [2046] [2052] [2044] [2056] [707] [698] [1294]
K300		6,0000	3,1438	8,0000	4,1917	[2722] [2720] [2721]
K301		4,0000	1,4080	4,0000	1,4080	[1052] [2248]
K302		6,0000	3,0752	6,0000	3,0752	[1511] [996] [2608]
K303		7,0000	3,1921	10,0000	4,5602	[148] [1254] [1070] [145] [2127]
K304		10,0000	4,4601	24,0000	10,7042	[1191] [1192] [1462] [1463]
K305		9,0000	3,3014	10,0000	3,6682	[550] [566] [756] [218] [593]
K306		8,0000	2,8872	8,0000	2,8872	[1748] [1519] [3] [101]
K307		6,0000	2,0800	4,0000	1,3867	[1698] [1956]
K308		6,0000	2,0800	4,0000	1,3867	[1699] [1957]
K309		4,0000	1,3771	4,0000	1,3771	[1189] [1811]
K310		37,0000	12,7280	33,0000	11,3520	[1580] [2270]
K311		30,0000	15,2297	102,0000	51,7809	[666] [1324] [320] [1336] [170] [1108] [803] [1162] [1790] [1831] [2555] [1155] [1384] [1234] [861] [440] [2260] [1813] [1260] [1092] [1202] [1576] [1895]
K312		122,0000	56,0071	199,0000	91,3558	[2829] [2822] [2747] [2814]
K313		12,0000	4,5559	18,0000	6,8338	[2167] [2913] [2327] [1631] [2335] [2266] [1389] [486] [821]
K314		22,0000	9,8205	36,0000	16,0699	[1184] [1185] [1444] [1445]
K315		7,0000	2,3569	5,0000	1,6835	[822] [1394]
K316		6,0000	2,1175	8,0000	2,8233	[49] [619] [778] [1724]
K317		8,0000	2,6667	6,0000	2,0000	[1488] [1492]
K318		28,0000	12,9028	21,0000	9,6771	[1262] [1267] [862] [1268] [1198] [1199]
K319		7,0000	2,2867	4,0000	1,3067	[1447] [1448]
K320		9,0000	4,0613	14,0000	6,3176	[430] [580] [739] [774]
K321		31,0000	12,6867	24,0000	9,8219	[3117] [3115] [3114] [3116] [3119] [3126]
K322		13,0000	8,0611	39,0000	24,1834	[75] [70] [1727] [620] [1807] [1783] [1800]
K323		4,0000	1,2923	4,0000	1,2923	[1560] [2617]
K324		8,0000	2,5778	5,0000	1,6111	[2132] [2136]
K325		10,0000	3,9253	14,0000	5,4954	[2541] [1008] [2064] [150] [2319] [1670] [1765]
K326		6,0000	2,2392	6,0000	2,2392	[2974] [19] [92]
K327		18,0000	6,6177	13,0000	4,7794	[2253] [2252] [2284] [2285]
K328		6,0000	1,8800	4,0000	1,2533	[2473] [2475]
K329		5,0000	1,5667	4,0000	1,2533	[2729] [2731]
K330		22,0000	6,8514	20,0000	6,2286	[22] [1752]
K331	Sim	58,0000	34,5235	172,0000	102,3800	[3008] [1925] [871] [815] [1385] [690] [1169] [1329] [1116] [1343] [472] [1047]
K332	Sim	58,0000	33,2845	172,0000	98,7059	[3009] [1926] [872] [1170] [691] [1330] [1117] [1344] [1048] [471] [816] [1386]
K333		15,0000	6,9532	17,0000	7,8803	[1891] [2162] [317] [1826]
K334	Sim	9,0000	3,3148	17,0000	6,2613	[1566] [2633] [2434] [2435] [1567] [2634]
K335		7,0000	2,1389	5,0000	1,5278	[2900] [2901]
K336		9,0000	2,7479	13,0000	3,9692	[971] [976]
K337		25,0000	9,9019	22,0000	8,7137	[2858] [2859] [2852] [2861]
K338		10,0000	3,0476	6,0000	1,8286	[239] [240]
K339		7,0000	2,1333	4,0000	1,2190	[1875] [1876]
K340		6,0000	2,5042	15,0000	6,2606	[807] [809] [448] [449] [1034] [1035]
K341		9,0000	2,7430	6,0000	1,8287	[249] [250] [251]
K342		62,0000	18,6827	44,0000	13,2587	[1856] [3081]
K343		12,0000	4,3531	14,0000	5,0786	[2395] [1527] [2573]
K344		7,0000	2,1000	4,0000	1,2000	[1994] [1995]
K345		16,0000	6,0652	24,0000	9,0978	[877] [2085] [2119] [1101] [709] [3094] [1644] [1312] [700] [2389]

Continua na próxima página ...



... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K346		12,0000	4,3267	16,0000	5,7690	[2407] [1539] [2587]
K347		8,0000	3,3616	10,0000	4,2019	[2643] [988] [2604]
K348		8,0000	2,3531	5,0000	1,4707	[2131] [2172]
K349		10,0000	3,6142	7,0000	2,5300	[2594] [2596] [2614]
K350		33,0000	14,0324	124,0000	52,7276	[1572] [1893] [439] [802] [2557] [1706] [2257] [1323] [1333] [1109] [1161] [1153] [1363] [316] [1828] [1226] [1231] [167] [667] [1261]
K351		24,0000	9,6239	77,0000	30,8767	[1575] [808] [1789] [447] [1033] [255] [864] [173] [1165] [2147] [319] [1830] [1055] [2258] [1233] [1277]
K352		6,0000	1,7394	4,0000	1,1596	[716] [884]
K353		14,0000	4,1945	8,0000	2,3969	[1443] [1067] [1149]
K354	Sim	50,0000	31,7851	463,0000	294,3304	[3075] [2841] [2976] [3143] [2315] [2461] [2066] [2058] [2530] [1173] [2240] [2287] [2504] [2007] [1143] [1437] [962] [1860] [2223] [2674]
K355		8,0000	3,0224	8,0000	3,0224	[1886] [2383] [1013] [1024]
K356		8,0000	2,9913	8,0000	2,9913	[1887] [2384] [1014] [1025]
K357		7,0000	1,9833	4,0000	1,1333	[1455] [1456]
K358		9,0000	2,7506	10,0000	3,0562	[1637] [2346] [2502] [2545]
K359		5,0000	1,3978	4,0000	1,1182	[453] [459]
K360		8,0000	3,3281	8,0000	3,3281	[1310] [1311] [1129] [1130]
K361	Sim	25,0000	10,1694	42,0000	17,0847	[1906] [1907] [1593] [1594]
K362		5,0000	1,3792	4,0000	1,1033	[1794] [1795]
K363		16,0000	6,0262	27,0000	10,1692	[2624] [2625] [1565] [2433] [1564] [2432]
K364		12,0000	4,9968	23,0000	9,5771	[1388] [1396] [818] [819] [2669] [2670]
K365		6,0000	1,6385	7,0000	1,9116	[27] [1471]
K366		5,0000	1,5835	6,0000	1,9003	[2922] [2919] [2926]
K367		4,0000	1,0819	4,0000	1,0819	[2340] [2341]

Na Tabela C.3 são apresentados os métodos do sistema Tomcat com o respectivo *fan-in*. Devido a limitação de espaço, foram listados somente os métodos com *fan-in* a partir de 5.

Tabela C.3: Lista de métodos do Tomcat

[Id]	Método(Parâmetros)	<i>fan-in</i>
10]	void catalina.ant.AbstractCatalinaTask.execute()	16
11]	void catalina.ant.AbstractCatalinaTask.execute(String)	14
28]	void catalina.ant.BaseRedirectorHelperTask.handleOutput(String)	9
30]	void catalina.ant.BaseRedirectorHelperTask.handleErrorOutput(String)	8
47]	void catalina.ant.DeployTask.execute()	16
55]	void catalina.ant.InstallTask.execute()	16
83]	void catalina.ant.JKStatusUpdateTask.execute()	16
202]	boolean catalina.ant.jmx.JMXAccessorTask.isEcho()	9
246]	void catalina.ant.JMXGetTask.execute()	16
250]	void catalina.ant.JMXQueryTask.execute()	16
258]	void catalina.ant.JMXSetTask.execute()	16
260]	void catalina.ant.ListTask.execute()	16
264]	void catalina.ant.ReloadTask.execute()	16
268]	void catalina.ant.RemoveTask.execute()	16
272]	void catalina.ant.ResourcesTask.execute()	16
274]	void catalina.ant.RolesTask.execute()	16
276]	void catalina.ant.ServerinfoTask.execute()	16
280]	void catalina.ant.SessionsTask.execute()	16
284]	void catalina.ant.StartTask.execute()	16
288]	void catalina.ant.StopTask.execute()	16
292]	void catalina.ant.UndeployTask.execute()	16
313]	void catalina.authenticator.AuthenticatorBase.invoke(catalina.connector.Request,catalina.connector.Response)	20
314]	void catalina.authenticator.AuthenticatorBase.associate(String,catalina.Session)	5
321]	void catalina.authenticator.AuthenticatorBase.addLifecycleListener(catalina.LifecycleListener)	7
322]	catalina.LifecycleListener[] catalina.authenticator.AuthenticatorBase.findLifecycleListeners()	5
324]	void catalina.authenticator.AuthenticatorBase.start()	22
325]	void catalina.authenticator.AuthenticatorBase.stop()	25
377]	void catalina.authenticator.SingleSignOn.addLifecycleListener(catalina.LifecycleListener)	7
378]	catalina.LifecycleListener[] catalina.authenticator.SingleSignOn.findLifecycleListeners()	5
380]	void catalina.authenticator.SingleSignOn.start()	21
381]	void catalina.authenticator.SingleSignOn.stop()	24
384]	void catalina.authenticator.SingleSignOn.invoke(catalina.connector.Request,catalina.connector.Response)	20
392]	catalina.authenticator.SingleSignOnEntry catalina.authenticator.SingleSignOn.lookup(String)	6
408]	void catalina.authenticator.SSLAuthenticator.start()	22
409]	void catalina.authenticator.SSLAuthenticator.stop()	25
415]	void catalina.cluster.CatalinaCluster.send(catalina.cluster.ClusterMessage,catalina.cluster.Member)	8
448]	boolean catalina.cluster.ClusterManager.isSendClusterDomainOnly()	5
489]	boolean catalina.cluster.ClusterSession.isPrimarySession()	5
790]	boolean catalina.cluster.session.DeltaManager.isSendClusterDomainOnly()	5
811]	catalina.Session catalina.cluster.session.DeltaManager.createSession(String)	6
815]	catalina.Session catalina.cluster.session.DeltaManager.createEmptySession()	5
822]	void catalina.cluster.session.DeltaManager.addLifecycleListener(catalina.LifecycleListener)	7
823]	catalina.LifecycleListener[] catalina.cluster.session.DeltaManager.findLifecycleListeners()	5
825]	void catalina.cluster.session.DeltaManager.start()	21
830]	void catalina.cluster.session.DeltaManager.stop()	24
857]	void catalina.cluster.session.DeltaRequest.addAction(int,int,String,Object)	5
881]	boolean catalina.cluster.session.DeltaSession.isPrimarySession()	5
905]	boolean catalina.cluster.session.DeltaSession.isValid()	25
907]	void catalina.cluster.session.DeltaSession.access()	9
908]	void catalina.cluster.session.DeltaSession.endAccess()	8
910]	void catalina.cluster.session.DeltaSession.expire()	10
922]	void catalina.cluster.session.DeltaSession.resetDeltaRequest()	8
981]	void catalina.cluster.session.JvmRouteBinderValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
991]	void catalina.cluster.session.JvmRouteBinderValve.addLifecycleListener(catalina.LifecycleListener)	7

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
992	catalogina.LifecycleListener[] catalogina.cluster.session.JvmRouteBinderValve.findLifecycleListeners()	5
994	void catalogina.cluster.session.JvmRouteBinderValve.start()	21
995	void catalogina.cluster.session.JvmRouteBinderValve.stop()	24
1018	void catalogina.cluster.session.ReplicatedSession.removeAttribute(String)	5
1024	void catalogina.cluster.session.ReplicatedSession.expire()	13
1026	void catalogina.cluster.session.ReplicatedSession.readObjectData(ObjectInputStream)	6
1027	void catalogina.cluster.session.ReplicatedSession.writeObjectData(ObjectOutputStream)	6
1114	catalogina.Session catalogina.cluster.session.SimpleTcpReplicationManager.createSession(String)	6
1121	void catalogina.cluster.session.SimpleTcpReplicationManager.start()	24
1122	void catalogina.cluster.session.SimpleTcpReplicationManager.stop()	27
1139	void catalogina.cluster.tcp.AsyncSocketSender.disconnect()	5
1240	boolean catalogina.cluster.tcp.DataSender.isConnected()	9
1262	void catalogina.cluster.tcp.DataSender.disconnect()	5
1299	void catalogina.cluster.tcp.FastAsyncSocketSender.disconnect()	5
1319	void catalogina.cluster.tcp.IDataSender.disconnect()	5
1321	boolean catalogina.cluster.tcp.IDataSender.isConnected()	9
1345	void catalogina.cluster.tcp.PooledSocketSender.disconnect()	5
1432	boolean catalogina.cluster.tcp.ReplicationValve.isDoProcessingStats()	5
1444	void catalogina.cluster.tcp.ReplicationValve.invoke(catalogina.connector.Request,catalogina.connector.Response)	20
1506	void catalogina.cluster.tcp.SimpleTcpCluster.transferProperty(String,Object)	6
1514	void catalogina.cluster.tcp.SimpleTcpCluster.addLifecycleListener(catalogina.LifecycleListener)	7
1515	catalogina.LifecycleListener[] catalogina.cluster.tcp.SimpleTcpCluster.findLifecycleListeners()	6
1518	void catalogina.cluster.tcp.SimpleTcpCluster.start()	21
1526	void catalogina.cluster.tcp.SimpleTcpCluster.stop()	24
1530	void catalogina.cluster.tcp.SimpleTcpCluster.send(catalogina.cluster.ClusterMessage,catalogina.cluster.Member)	8
1744	void catalogina.connector.Connector.addLifecycleListener(catalogina.LifecycleListener)	7
1745	catalogina.LifecycleListener[] catalogina.connector.Connector.findLifecycleListeners()	5
1748	void catalogina.connector.Connector.initialize()	5
1751	void catalogina.connector.Connector.start()	21
1752	void catalogina.connector.Connector.stop()	25
1819	void catalogina.connector.CoyoteWriter.write(String)	7
1829	void catalogina.connector.CoyoteWriter.println()	9
1848	int catalogina.connector.InputBuffer.read(byte,int,int)	5
1887	void catalogina.connector.OutputBuffer.write(byte,int,int)	5
1974	boolean catalogina.connector.Request.isSecure()	7
2024	boolean catalogina.connector.Request.isRequestedSessionIdFromCookie()	7
2025	boolean catalogina.connector.Request.isRequestedSessionIdFromURL()	5
2157	boolean catalogina.connector.Response.isCommitted()	18
2174	void catalogina.connector.Response.addHeader(String,String)	5
2182	void catalogina.connector.Response.sendError(int)	6
2183	void catalogina.connector.Response.sendError(int,String)	10
2213	boolean catalogina.connector.ResponseFacade.isCommitted()	18
2262	void catalogina.Container.addChild(catalogina.Container)	11
2264	void catalogina.Container.addPropertyChangeListener(PropertyChangeListener)	8
2265	catalogina.Container catalogina.Container.findChild(String)	24
2266	catalogina.Container[] catalogina.Container.findChildren()	20
2269	void catalogina.Container.removeChild(catalogina.Container)	13
2271	void catalogina.Container.removePropertyChangeListener(PropertyChangeListener)	7
2349	catalogina.deploy.SecurityConstraint[] catalogina.Context.findConstraints()	5
2463	Object catalogina.core.ApplicationContextFacade.doPrivileged(catalogina.core.ApplicationContext,String,Object)	9
2464	Object catalogina.core.ApplicationContextFacade.doPrivileged(String,Object)	12
2617	void catalogina.core.ContainerBase.addChild(catalogina.Container)	11
2620	void catalogina.core.ContainerBase.addPropertyChangeListener(PropertyChangeListener)	8
2621	catalogina.Container catalogina.core.ContainerBase.findChild(String)	24
2622	catalogina.Container[] catalogina.core.ContainerBase.findChildren()	20
2625	void catalogina.core.ContainerBase.removeChild(catalogina.Container)	13
2627	void catalogina.core.ContainerBase.removePropertyChangeListener(PropertyChangeListener)	7
2628	void catalogina.core.ContainerBase.addLifecycleListener(catalogina.LifecycleListener)	13
2629	catalogina.LifecycleListener[] catalogina.core.ContainerBase.findLifecycleListeners()	6
2631	void catalogina.core.ContainerBase.start()	27
2632	void catalogina.core.ContainerBase.stop()	31
2633	void catalogina.core.ContainerBase.init()	5
2635	void catalogina.core.ContainerBase.destroy()	6
2636	void catalogina.core.ContainerBase.addValve(catalogina.Valve)	8
2644	void catalogina.core.ContainerBase.fireContainerEvent(String,Object)	55
2645	String catalogina.core.ContainerBase.logName()	6
2872	void catalogina.core.NamingContextListener.createSubcontexts(Context,String)	5
2977	void catalogina.core.StandardContext.addChild(catalogina.Container)	11
3001	catalogina.deploy.SecurityConstraint[] catalogina.core.StandardContext.findConstraints()	5
3035	void catalogina.core.StandardContext.removeChild(catalogina.Container)	13
3062	void catalogina.core.StandardContext.start()	27
3065	void catalogina.core.StandardContext.stop()	31
3066	void catalogina.core.StandardContext.destroy()	5
3127	void catalogina.core.StandardContextValve.invoke(catalogina.connector.Request,catalogina.connector.Response)	20
3144	void catalogina.core.StandardEngine.addChild(catalogina.Container)	11
3149	void catalogina.core.StandardEngine.start()	24
3150	void catalogina.core.StandardEngine.stop()	27
3160	void catalogina.core.StandardEngineValve.invoke(catalogina.connector.Request,catalogina.connector.Response)	20
3189	void catalogina.core.StandardHost.addChild(catalogina.Container)	11
3195	void catalogina.core.StandardHost.start()	24
3204	void catalogina.core.StandardHostValve.invoke(catalogina.connector.Request,catalogina.connector.Response)	20
3214	void catalogina.core.StandardPipeline.addLifecycleListener(catalogina.LifecycleListener)	7
3215	catalogina.LifecycleListener[] catalogina.core.StandardPipeline.findLifecycleListeners()	5
3217	void catalogina.core.StandardPipeline.start()	21
3218	void catalogina.core.StandardPipeline.stop()	24
3252	void catalogina.core.StandardServer.addLifecycleListener(catalogina.LifecycleListener)	8
3253	catalogina.LifecycleListener[] catalogina.core.StandardServer.findLifecycleListeners()	5
3255	void catalogina.core.StandardServer.start()	21
3256	void catalogina.core.StandardServer.stop()	24
3281	void catalogina.core.StandardService.addLifecycleListener(catalogina.LifecycleListener)	7
3282	catalogina.LifecycleListener[] catalogina.core.StandardService.findLifecycleListeners()	5

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
3284	void catalina.core.StandardService.start()	22
3285	void catalina.core.StandardService.stop()	27
3320	void catalina.core.StandardWrapper.addChild(catalina.Container)	11
3364	void catalina.core.StandardWrapper.start()	24
3365	void catalina.core.StandardWrapper.stop()	27
3380	void catalina.core.StandardWrapperValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
3542	catalina.deploy.ContextEnvironment catalina.deploy.NamingResources.findEnvironment(String)	7
3543	catalina.deploy.ContextEnvironment[] catalina.deploy.NamingResources.findEnvironments()	7
3548	catalina.deploy.ContextResource catalina.deploy.NamingResources.findResource(String)	7
3549	catalina.deploy.ContextResourceLink catalina.deploy.NamingResources.findResourceLink(String)	7
3550	catalina.deploy.ContextResourceLink[] catalina.deploy.NamingResources.findResourceLinks()	7
3551	catalina.deploy.ContextResource[] catalina.deploy.NamingResources.findResources()	7
3633	void catalina.ha.CatalinaCluster.send(catalina.ha.ClusterMessage,catalina.tribes.Member)	8
3674	boolean catalina.ha.ClusterManager.isSendClusterDomainOnly()	5
3697	boolean catalina.ha.ClusterSession.isPrimarySession()	5
3703	void catalina.ha.context.ReplicatedContext.start()	27
3704	void catalina.ha.context.ReplicatedContext.stop()	31
3829	void catalina.ha.session.BackupManager.start()	24
3831	void catalina.ha.session.BackupManager.stop()	27
3880	boolean catalina.ha.session.DeltaManager.isSendClusterDomainOnly()	5
3901	catalina.Session catalina.ha.session.DeltaManager.createSession(String)	6
3905	catalina.Session catalina.ha.session.DeltaManager.createEmptySession()	5
3911	void catalina.ha.session.DeltaManager.addLifecycleListener(catalina.LifecycleListener)	7
3912	catalina.LifecycleListener[] catalina.ha.session.DeltaManager.findLifecycleListeners()	5
3914	void catalina.ha.session.DeltaManager.start()	21
3919	void catalina.ha.session.DeltaManager.stop()	24
3946	void catalina.ha.session.DeltaRequest.addAction(int,int,String,Object)	5
3977	void catalina.ha.session.DeltaSession.lock()	5
3978	void catalina.ha.session.DeltaSession.unlock()	5
3980	boolean catalina.ha.session.DeltaSession.isPrimarySession()	5
4004	boolean catalina.ha.session.DeltaSession.isValid()	25
4006	void catalina.ha.session.DeltaSession.access()	10
4007	void catalina.ha.session.DeltaSession.endAccess()	9
4009	void catalina.ha.session.DeltaSession.expire()	10
4022	void catalina.ha.session.DeltaSession.resetDeltaRequest()	10
4082	void catalina.ha.session.JvmRouteBinderValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
4092	void catalina.ha.session.JvmRouteBinderValve.addLifecycleListener(catalina.LifecycleListener)	7
4093	catalina.LifecycleListener[] catalina.ha.session.JvmRouteBinderValve.findLifecycleListeners()	5
4095	void catalina.ha.session.JvmRouteBinderValve.start()	21
4096	void catalina.ha.session.JvmRouteBinderValve.stop()	24
4119	void catalina.ha.session.ReplicatedSession.removeAttribute(String)	5
4125	void catalina.ha.session.ReplicatedSession.expire()	13
4127	void catalina.ha.session.ReplicatedSession.readObjectData(ObjectInputStream)	6
4128	void catalina.ha.session.ReplicatedSession.writeObjectData(ObjectOutputStream)	6
4197	catalina.Session catalina.ha.session.SimpleTcpReplicationManager.createSession(String)	6
4206	void catalina.ha.session.SimpleTcpReplicationManager.start()	24
4207	void catalina.ha.session.SimpleTcpReplicationManager.stop()	27
4228	boolean catalina.ha.tcp.ReplicationValve.isDoProcessingStats()	5
4240	void catalina.ha.tcp.ReplicationValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
4296	void catalina.ha.tcp.SimpleTcpCluster.addLifecycleListener(catalina.LifecycleListener)	7
4297	catalina.LifecycleListener[] catalina.ha.tcp.SimpleTcpCluster.findLifecycleListeners()	5
4300	void catalina.ha.tcp.SimpleTcpCluster.start()	21
4303	void catalina.ha.tcp.SimpleTcpCluster.stop()	24
4306	void catalina.ha.tcp.SimpleTcpCluster.send(catalina.ha.ClusterMessage,catalina.tribes.Member)	8
4355	void catalina.InstanceListener.instanceEvent(catalina.InstanceEvent)	8
4358	void catalina.Lifecycle.addLifecycleListener(catalina.LifecycleListener)	14
4359	catalina.LifecycleListener[] catalina.Lifecycle.findLifecycleListeners()	7
4361	void catalina.Lifecycle.start()	39
4362	void catalina.Lifecycle.stop()	46
4415	void catalina.loader.WebappClassLoader.addLifecycleListener(catalina.LifecycleListener)	7
4416	catalina.LifecycleListener[] catalina.loader.WebappClassLoader.findLifecycleListeners()	5
4418	void catalina.loader.WebappClassLoader.start()	21
4419	void catalina.loader.WebappClassLoader.stop()	24
4463	void catalina.loader.WebappLoader.addLifecycleListener(catalina.LifecycleListener)	7
4464	catalina.LifecycleListener[] catalina.loader.WebappLoader.findLifecycleListeners()	5
4468	void catalina.loader.WebappLoader.start()	21
4469	void catalina.loader.WebappLoader.stop()	24
4520	void catalina.Manager.add(catalina.Session)	11
4522	catalina.Session catalina.Manager.createEmptySession()	7
4524	catalina.Session catalina.Manager.createSession(String)	6
4525	catalina.Session catalina.Manager.findSession(String)	29
4526	catalina.Session[] catalina.Manager.findSessions()	16
4528	void catalina.Manager.remove(catalina.Session)	13
4603	String catalina.mbeans.MBeanUtils.createManagedName(Object)	38
4623	ObjectName catalina.mbeans.MBeanUtils.createObjectName(String,catalina.connector.Connector)	65
4644	MBeanServer catalina.mbeans.MBeanUtils.createServer()	5
4753	void catalina.Pipeline.addValue(catalina.Valve)	8
4785	void catalina.realm.DataSourceRealm.start()	27
4786	void catalina.realm.DataSourceRealm.stop()	30
4824	void catalina.realm.JAASRealm.start()	27
4825	void catalina.realm.JAASRealm.stop()	30
4856	void catalina.realm.JDBCRealm.start()	27
4857	void catalina.realm.JDBCRealm.stop()	30
4916	void catalina.realm.JNDIRealm.start()	27
4917	void catalina.realm.JNDIRealm.stop()	30
4932	void catalina.realm.MemoryRealm.start()	27
4933	void catalina.realm.MemoryRealm.stop()	30
4963	void catalina.realm.RealmBase.addLifecycleListener(catalina.LifecycleListener)	7
4964	catalina.LifecycleListener[] catalina.realm.RealmBase.findLifecycleListeners()	5
4966	void catalina.realm.RealmBase.start()	27
4967	void catalina.realm.RealmBase.stop()	30

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
4969	String catalina.realm.RealmBase.digest(String)	6
4970	boolean catalina.realm.RealmBase.hasMessageDigest()	8
5001	void catalina.realm.UserDatabaseRealm.start()	27
5002	void catalina.realm.UserDatabaseRealm.stop()	30
5038	void catalina.security.SecurityUtil.doAsPrivilege(String,Servlet,Class,Object)	5
5046	boolean catalina.security.SecurityUtil.isPackageProtectionEnabled()	46
5190	boolean catalina.servlets.WebdavServlet.isLocked(http.HttpServletRequest)	7
5223	void catalina.session.FileStore.remove(String)	5
5257	void catalina.session.JDBCStore.remove(String)	5
5262	void catalina.session.JDBCStore.close(Connection)	7
5263	void catalina.session.JDBCStore.release(Connection)	6
5264	void catalina.session.JDBCStore.start()	22
5265	void catalina.session.JDBCStore.stop()	25
5300	void catalina.session.ManagerBase.add(catalina.Session)	11
5303	catalina.Session catalina.session.ManagerBase.createSession(String)	6
5304	catalina.Session catalina.session.ManagerBase.createEmptySession()	7
5305	catalina.Session catalina.session.ManagerBase.findSession(String)	29
5306	catalina.Session[] catalina.session.ManagerBase.findSessions()	16
5307	void catalina.session.ManagerBase.remove(catalina.Session)	13
5311	String catalina.session.ManagerBase.generateSessionId()	7
5373	catalina.Session catalina.session.PersistentManagerBase.findSession(String)	29
5376	void catalina.session.PersistentManagerBase.remove(catalina.Session)	13
5382	void catalina.session.PersistentManagerBase.addLifecycleListener(catalina.LifecycleListener)	7
5383	catalina.LifecycleListener[] catalina.session.PersistentManagerBase.findLifecycleListeners()	5
5385	void catalina.session.PersistentManagerBase.start()	21
5386	void catalina.session.PersistentManagerBase.stop()	24
5405	catalina.Session catalina.session.StandardManager.createSession(String)	6
5410	void catalina.session.StandardManager.addLifecycleListener(catalina.LifecycleListener)	7
5411	catalina.LifecycleListener[] catalina.session.StandardManager.findLifecycleListeners()	5
5413	void catalina.session.StandardManager.start()	24
5414	void catalina.session.StandardManager.stop()	27
5436	boolean catalina.session.StandardSession.isValid()	30
5438	void catalina.session.StandardSession.access()	5
5439	void catalina.session.StandardSession.endAccess()	6
5441	void catalina.session.StandardSession.expire()	13
5442	void catalina.session.StandardSession.expire(boolean)	5
5447	void catalina.session.StandardSession.recycle()	5
5452	void catalina.session.StandardSession.readObjectData(ObjectInputStream)	7
5453	void catalina.session.StandardSession.writeObjectData(ObjectOutputStream)	7
5464	void catalina.session.StandardSession.removeAttribute(String)	6
5474	String[] catalina.session.StandardSession.keys()	5
5503	void catalina.session.StoreBase.addLifecycleListener(catalina.LifecycleListener)	7
5504	catalina.LifecycleListener[] catalina.session.StoreBase.findLifecycleListeners()	5
5509	void catalina.session.StoreBase.start()	22
5510	void catalina.session.StoreBase.stop()	25
5530	boolean catalina.Session.isValid()	52
5531	void catalina.Session.access()	14
5533	void catalina.Session.endAccess()	13
5534	void catalina.Session.expire()	19
5537	void catalina.Session.recycle()	5
5661	String catalina.ssi.SSIMediator.substituteVariables(String)	7
5664	void catalina.ssi.SSIMediator.log(String)	6
5756	void catalina.startup.Catalina.start()	22
5757	void catalina.startup.Catalina.stop()	27
5828	tomcat.util.digester.Digester catalina.startup.DigesterFactory.newDigester(boolean,boolean,tomcat.util.digester.RuleSet)	5
5861	void catalina.startup.Embedded.addLifecycleListener(catalina.LifecycleListener)	7
5862	catalina.LifecycleListener[] catalina.startup.Embedded.findLifecycleListeners()	5
5864	void catalina.startup.Embedded.start()	22
5865	void catalina.startup.Embedded.stop()	27
5905	boolean catalina.startup.HostConfig.isServiced(String)	5
5909	File catalina.startup.HostConfig.appBase()	7
6029	void catalina.Store.remove(String)	6
6047	boolean catalina.storeconfig.ConnectorStoreAppender.isPrintValue(Object,Object,String,StoreDescription)	5
6052	void catalina.storeconfig.GlobalNamingResourcesSF.store(PrintWriter,int,Object)	14
6057	void catalina.storeconfig.InstanceListenerSF.store(PrintWriter,int,Object)	14
6071	void catalina.storeconfig.IStoreFactory.store(PrintWriter,int,Object)	14
6074	void catalina.storeconfig.LoaderSF.store(PrintWriter,int,Object)	14
6077	void catalina.storeconfig.ManagerSF.store(PrintWriter,int,Object)	14
6080	void catalina.storeconfig.NamingResourcesSF.store(PrintWriter,int,Object)	14
6085	void catalina.storeconfig.StandardContextSF.store(PrintWriter,int,Object)	14
6097	void catalina.storeconfig.StandardServerSF.store(PrintWriter,int,Object)	14
6107	void catalina.storeconfig.StoreAppender.printTagArray(PrintWriter,String,int,String)	6
6108	void catalina.storeconfig.StoreAppender.printIndent(PrintWriter,int)	7
6111	boolean catalina.storeconfig.StoreAppender.isPrintValue(Object,Object,String,StoreDescription)	5
6115	boolean catalina.storeconfig.StoreAppender.isPersistable(Class)	5
6179	boolean catalina.storeconfig.StoreDescription.isStoreSeparate()	6
6190	void catalina.storeconfig.StoreFactoryBase.store(PrintWriter,int,Object)	14
6192	void catalina.storeconfig.StoreFactoryBase.storeElement(PrintWriter,int,Object)	6
6193	void catalina.storeconfig.StoreFactoryBase.storeElementArray(PrintWriter,int,Object)	8
6228	catalina.storeconfig.StoreDescription catalina.storeconfig.StoreRegistry.findDescription(String)	8
6229	catalina.storeconfig.StoreDescription catalina.storeconfig.StoreRegistry.findDescription(Class)	14
6237	void catalina.storeconfig.WatchedResourceSF.store(PrintWriter,int,Object)	14
6239	void catalina.storeconfig.WrapperLifecycleSF.store(PrintWriter,int,Object)	14
6241	void catalina.storeconfig.WrapperListenerSF.store(PrintWriter,int,Object)	14
6251	void catalina.tribes.Channel.send(catalina.tribes.Member,Serializable,int)	10
6267	void catalina.tribes.ChannelException.addFaultyMember(catalina.tribes.Member)	11
6278	void catalina.tribes.ChannelInterceptor.sendMessage(Member,ChannelMessage,InterceptorPayload)	8
6279	void catalina.tribes.ChannelInterceptor.messageReceived(catalina.tribes.ChannelMessage)	6
6319	void catalina.tribes.group.ChannelCoordinator.sendMessage(Member,ChannelMessage,InterceptorPayload)	8
6324	void catalina.tribes.group.ChannelCoordinator.memberAdded(catalina.tribes.Member)	6
6325	void catalina.tribes.group.ChannelCoordinator.memberDisappeared(catalina.tribes.Member)	6

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
6326	void catalina.tribes.group.ChannelCoordinator.messageReceived(catalina.tribes.ChannelMessage)	7
6345	void catalina.tribes.group.ChannelInterceptorBase.sendMessage(Member,ChannelMessage,InterceptorPayload)	8
6346	void catalina.tribes.group.ChannelInterceptorBase.messageReceived(catalina.tribes.ChannelMessage)	6
6348	void catalina.tribes.group.ChannelInterceptorBase.memberAdded(catalina.tribes.Member)	6
6349	void catalina.tribes.group.ChannelInterceptorBase.memberDisappeared(catalina.tribes.Member)	6
6359	void catalina.tribes.group.GroupChannel.heartbeat()	5
6360	void catalina.tribes.group.GroupChannel.send(catalina.tribes.Member,Serializable,int)	10
6362	void catalina.tribes.group.GroupChannel.messageReceived(catalina.tribes.ChannelMessage)	6
6363	void catalina.tribes.group.GroupChannel.memberAdded(catalina.tribes.Member)	6
6364	void catalina.tribes.group.GroupChannel.memberDisappeared(catalina.tribes.Member)	6
6392	void catalina.tribes.group.interceptors.FragmentationInterceptor.sendMessage(Member,ChannelMessage,InterceptorPayload)	8
6393	void catalina.tribes.group.interceptors.FragmentationInterceptor.messageReceived(catalina.tribes.ChannelMessage)	6
6413	void catalina.tribes.group.interceptors.GzipInterceptor.sendMessage(Member,ChannelMessage,InterceptorPayload)	8
6414	void catalina.tribes.group.interceptors.GzipInterceptor.messageReceived(catalina.tribes.ChannelMessage)	6
6419	void catalina.tribes.group.interceptors.MessageDispatchInterceptor.sendMessage(Member,ChannelMessage,InterceptorPayload)	8
6428	void catalina.tribes.group.interceptors.OrderInterceptor.sendMessage(Member,ChannelMessage,InterceptorPayload)	8
6429	void catalina.tribes.group.interceptors.OrderInterceptor.messageReceived(catalina.tribes.ChannelMessage)	6
6432	void catalina.tribes.group.interceptors.OrderInterceptor.memberAdded(catalina.tribes.Member)	6
6433	void catalina.tribes.group.interceptors.OrderInterceptor.memberDisappeared(catalina.tribes.Member)	6
6526	void catalina.tribes.io.XByteBuffer.trim(int)	6
6535	boolean catalina.tribes.io.XByteBuffer.append(byte,int,int)	6
6536	void catalina.tribes.io.XByteBuffer.expand(int)	7
6538	int catalina.tribes.io.XByteBuffer.countPackages()	6
6547	int catalina.tribes.io.XByteBuffer.toInt(byte,int)	9
6550	byte[] catalina.tribes.io.XByteBuffer.toByteArray(boolean,byte,int)	9
6678	void catalina.tribes.MembershipListener.memberAdded(catalina.tribes.Member)	6
6679	void catalina.tribes.MembershipListener.memberDisappeared(catalina.tribes.Member)	7
6717	void catalina.tribes.tipsis.AbstractReplicatedMap.memberDisappeared(catalina.tribes.Member)	5
6719	catalina.tribes.Member[] catalina.tribes.tipsis.AbstractReplicatedMap.publishEntryInfo(Object,Object)	5
6725	boolean catalina.tribes.tipsis.AbstractReplicatedMap\$MapEntry.isPrimary()	12
6817	void catalina.tribes.tipsis.ReplicatedMapEntry.lock()	5
6818	void catalina.tribes.tipsis.ReplicatedMapEntry.unlock()	5
6834	void catalina.tribes.transport.AbstractSender.connect()	5
6835	void catalina.tribes.transport.AbstractSender.disconnect()	13
6836	boolean catalina.tribes.transport.AbstractSender.keepalive()	7
6838	boolean catalina.tribes.transport.AbstractSender.isConnected()	12
6895	void catalina.tribes.transport.bio.BioSender.disconnect()	6
6909	void catalina.tribes.transport.bio.MultipointBioSender.disconnect()	5
6911	boolean catalina.tribes.transport.bio.MultipointBioSender.keepalive()	7
6962	void catalina.tribes.transport.DataSender.connect()	5
6963	void catalina.tribes.transport.DataSender.disconnect()	13
6964	boolean catalina.tribes.transport.DataSender.isConnected()	12
6967	boolean catalina.tribes.transport.DataSender.keepalive()	7
7004	void catalina.tribes.transport.nio.NioSender.disconnect()	7
7026	void catalina.tribes.transport.nio.ParallelNioSender.disconnect()	6
7028	boolean catalina.tribes.transport.nio.ParallelNioSender.keepalive()	7
7032	void catalina.tribes.transport.nio.PooledParallelSender.disconnect()	5
7041	void catalina.tribes.transport.PooledSender.disconnect()	5
7046	boolean catalina.tribes.transport.PooledSender.keepalive()	7
7193	catalina.Group catalina.UserDatabase.findGroup(String)	5
7194	catalina.Role catalina.UserDatabase.findRole(String)	9
7279	catalina.Group catalina.users.MemoryUserDatabase.findGroup(String)	5
7280	catalina.Role catalina.users.MemoryUserDatabase.findRole(String)	9
7406	void catalina.util.LifecycleSupport.addLifecycleListener(catalina.LifecycleListener)	24
7407	catalina.LifecycleListener[] catalina.util.LifecycleSupport.findLifecycleListeners()	24
7408	void catalina.util.LifecycleSupport.fireLifecycleEvent(String,Object)	68
7409	void catalina.util.LifecycleSupport.removeLifecycleListener(catalina.LifecycleListener)	24
7423	String catalina.util.MD5Encoder.encode(byte)	6
7480	String catalina.util.RequestUtil.URLDecode(String)	8
7568	void catalina.util.XMLWriter.writeElement(String,String,int)	9
7570	void catalina.util.XMLWriter.writeText(String)	5
7578	void catalina.Valve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7597	void catalina.valves.AccessLogValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7607	void catalina.valves.AccessLogValve.addLifecycleListener(catalina.LifecycleListener)	7
7608	catalina.LifecycleListener[] catalina.valves.AccessLogValve.findLifecycleListeners()	5
7610	void catalina.valves.AccessLogValve.start()	21
7611	void catalina.valves.AccessLogValve.stop()	24
7616	void catalina.valves.ErrorReportValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7636	void catalina.valves.ExtendedAccessLogValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7647	void catalina.valves.ExtendedAccessLogValve.addLifecycleListener(catalina.LifecycleListener)	7
7648	catalina.LifecycleListener[] catalina.valves.ExtendedAccessLogValve.findLifecycleListeners()	5
7650	void catalina.valves.ExtendedAccessLogValve.start()	21
7651	void catalina.valves.ExtendedAccessLogValve.stop()	24
7675	void catalina.valves.FastCommonAccessLogValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7683	void catalina.valves.FastCommonAccessLogValve.addLifecycleListener(catalina.LifecycleListener)	7
7684	catalina.LifecycleListener[] catalina.valves.FastCommonAccessLogValve.findLifecycleListeners()	5
7686	void catalina.valves.FastCommonAccessLogValve.start()	21
7687	void catalina.valves.FastCommonAccessLogValve.stop()	24
7708	void catalina.valves.JDBCAccessLogValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7709	void catalina.valves.JDBCAccessLogValve.addLifecycleListener(catalina.LifecycleListener)	7
7710	catalina.LifecycleListener[] catalina.valves.JDBCAccessLogValve.findLifecycleListeners()	5
7714	void catalina.valves.JDBCAccessLogValve.start()	21
7715	void catalina.valves.JDBCAccessLogValve.stop()	24
7719	void catalina.valves.PersistentValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7723	void catalina.valves.RemoteAddrValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7726	void catalina.valves.RemoteHostValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7729	void catalina.valves.RequestDumperValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7737	void catalina.valves.RequestFilterValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7749	void catalina.valves.SemaphoreValve.addLifecycleListener(catalina.LifecycleListener)	7
7750	catalina.LifecycleListener[] catalina.valves.SemaphoreValve.findLifecycleListeners()	5
7752	void catalina.valves.SemaphoreValve.start()	21

Continua na próxima página ...

... continuação da página anterior		
[Id] Método(Parâmetros)	fan-in	
7753	void catalina.valves.SemaphoreValve.stop()	24
7755	void catalina.valves.SemaphoreValve.invoke(catalina.connector.Request,catalina.connector.Response)	20
7765	void catalina.valves.ValveBase.invoke(catalina.connector.Request,catalina.connector.Response)	20
7867	void JasperCompiler.Dumper\$DumpVisitor.printString(String)	11
7868	void JasperCompiler.Dumper\$DumpVisitor.printString(String,char,String)	6
7870	void JasperCompiler.Dumper\$DumpVisitor.printAttributes(String,Attributes,String)	13
7871	void JasperCompiler.Dumper\$DumpVisitor.dumpBody(JasperCompiler.Node)	12
7909	void JasperCompiler.ELFunctionMapper\$ELFunctionVisitor.doMap(JasperCompiler.Node\$JspAttribute)	9
7959	boolean JasperCompiler.ELParser.hasNextChar()	6
7975	void JasperCompiler.ErrorDispatcher.jspError(String)	5
7976	void JasperCompiler.ErrorDispatcher.jspError(JasperCompiler.Mark,String)	54
7979	void JasperCompiler.ErrorDispatcher.jspError(JasperCompiler.Mark,String,String)	46
7982	void JasperCompiler.ErrorDispatcher.jspError(String,String,String,String)	14
7993	void JasperCompiler.ErrorDispatcher.dispatch(JasperCompiler.Mark,String,Object,Exception)	15
7997	void JasperCompiler.ErrorDispatcher\$ErrorVisitor.doVisit(JasperCompiler.Node)	33
8003	String JasperCompiler.Generator.quote(String)	18
8020	String JasperCompiler.Generator\$GenerateVisitor.attributeValue(JasperCompiler.Node\$JspAttribute,boolean,Class)	7
8059	String JasperCompiler.Generator\$GenerateVisitor.generateNamedAttributeValue(JasperCompiler.Node\$NamedAttribute)	8
8132	void JasperCompiler.JspDocumentParser.processChars()	5
8161	boolean JasperCompiler.JspReader.hasMoreInput()	9
8162	int JasperCompiler.JspReader.nextChar()	18
8165	int JasperCompiler.JspReader.peekChar()	8
8166	JasperCompiler.Mark JasperCompiler.JspReader.mark()	42
8167	void JasperCompiler.JspReader.reset(JasperCompiler.Mark)	12
8169	boolean JasperCompiler.JspReader.matches(String)	24
8170	boolean JasperCompiler.JspReader.matchesETag(String)	5
8173	int JasperCompiler.JspReader.skipSpaces()	30
8174	JasperCompiler.Mark JasperCompiler.JspReader.skipUntil(String)	11
8213	void JasperCompiler.JspUtil.checkAttributes(String,Node,JspUtil\$ValidAttribute,ErrorDispatcher)	18
8220	boolean JasperCompiler.JspUtil.getBooleanValue(String)	8
8306	void JasperCompiler.Node.addToParent(JasperCompiler.Node)	5
8309	boolean JasperCompiler.Node\$Root.isXmlSyntax()	9
8513	boolean JasperCompiler.Node\$JspAttribute.isExpression()	12
8514	boolean JasperCompiler.Node\$JspAttribute.isNamedAttribute()	8
8521	void JasperCompiler.Node\$Nodes.add(JasperCompiler.Node)	5
8523	void JasperCompiler.Node\$Nodes.visit(JasperCompiler.Node\$Visitor)	26
8524	int JasperCompiler.Node\$Nodes.size()	18
8530	void JasperCompiler.Node\$Visitor.doVisit(JasperCompiler.Node)	33
8531	void JasperCompiler.Node\$Visitor.visitBody(JasperCompiler.Node)	86
8604	void JasperCompiler.PageDataImpl\$SecondPassVisitor.appendTag(JasperCompiler.Node)	22
8689	Attributes JasperCompiler.Parser.parseAttributes()	19
8695	String JasperCompiler.Parser.parseScriptText(String)	6
8724	void JasperCompiler.Parser.parseOptionalBody(JasperCompiler.Node,String,String)	10
8768	void JasperCompiler.ServletWriter.pushIndent()	24
8769	void JasperCompiler.ServletWriter.popIndent()	25
8771	void JasperCompiler.ServletWriter.println(String)	35
8772	void JasperCompiler.ServletWriter.println()	16
8774	void JasperCompiler.ServletWriter.println(String)	35
8775	void JasperCompiler.ServletWriter.print(String)	34
8778	void JasperCompiler.ServletWriter.print(String)	33
8822	int JasperCompiler.SmapUtil\$SDEInstaller.readU2()	5
8825	void JasperCompiler.SmapUtil\$SDEInstaller.writeU2(int)	8
8834	void JasperCompiler.SmapUtil\$SmapGenVisitor.visitBody(JasperCompiler.Node)	86
8855	void JasperCompiler.SmapUtil\$SmapGenVisitor.doSmap(JasperCompiler.Node)	14
8858	void JasperCompiler.SmapUtil\$PreScanVisitor.doVisit(JasperCompiler.Node)	33
8900	boolean JasperCompiler.tagplugin.TagPluginContext.isAttributeSpecified(String)	10
8904	void JasperCompiler.tagplugin.TagPluginContext.generateJavaSource(String)	15
8907	void JasperCompiler.tagplugin.TagPluginContext.generateAttribute(String)	11
8908	void JasperCompiler.tagplugin.TagPluginContext.generateBody()	12
8924	boolean JasperCompiler.TagPluginManager\$TagPluginContextImpl.isAttributeSpecified(String)	10
8928	void JasperCompiler.TagPluginManager\$TagPluginContextImpl.generateJavaSource(String)	15
8929	void JasperCompiler.TagPluginManager\$TagPluginContextImpl.generateAttribute(String)	11
8931	void JasperCompiler.TagPluginManager\$TagPluginContextImpl.generateBody()	12
8935	void JasperCompiler.TextOptimizer\$TextCatVisitor.doVisit(JasperCompiler.Node)	33
8941	void JasperCompiler.TextOptimizer\$TextCatVisitor.visitBody(JasperCompiler.Node)	72
8995	void JasperCompiler.Validator\$ValidateVisitor\$NamedAttributeVisitor.doVisit(JasperCompiler.Node)	33
9146	boolean Jasper.JspCompilationContext.isTagFile()	9
9149	boolean Jasper.JspCompilationContext.isPrototypeMode()	5
9218	void Jasper.runtime.BodyContentImpl.println()	9
9353	void Jasper.runtime.JspWriterImpl.flushBuffer()	5
9362	void Jasper.runtime.JspWriterImpl.ensureOpen()	6
9368	void Jasper.runtime.JspWriterImpl.write(String)	9
9379	void Jasper.runtime.JspWriterImpl.println()	9
9463	boolean Jasper.security.SecurityUtil.isPackageProtectionEnabled()	17
9590	PrintStream Jasper.util.SystemLogHandler.findStream()	25
9634	Jasper.xmlparser.TreeNode Jasper.xmlparser.ParserUtils.parseXMLDocument(String,InputSource)	6
9659	Jasper.xmlparser.TreeNode Jasper.xmlparser.TreeNode.findChild(String)	5
9660	Iterator Jasper.xmlparser.TreeNode.findChildren()	10
9720	boolean Jasper.xmlparser.XMLEncodingDetector.isExternal()	6
9721	int Jasper.xmlparser.XMLEncodingDetector.peekChar()	5
9729	boolean Jasper.xmlparser.XMLEncodingDetector.load(int,boolean)	9
9772	boolean naming.ContextAccessController.checkSecurityToken(Object,Object)	8
9978	void naming.resources.DirContextURLConnection.connect()	5
10035	File naming.resources.FileDirContext.file(String)	6
10038	InputStream naming.resources.FileDirContext\$fileResource.streamContent()	19
10063	Object naming.resources.ProxyDirContext.lookup(String)	7
10118	String naming.resources.ProxyDirContext.parseName(String)	25
10119	Name naming.resources.ProxyDirContext.parseName(Name)	24
10120	naming.resources.CacheEntry naming.resources.ProxyDirContext.cacheLookup(String)	5
10124	boolean naming.resources.ProxyDirContext.cacheUnload(String)	23
10135	InputStream naming.resources.Resource.streamContent()	19

Continua na próxima página ...

... continuação da página anterior	
[Id] Método(Parâmetros)	fan-in
[10225]naming.resources.WARDirContext.Entry naming.resources.WARDirContext.treeLookup(Name)	5
[10235]InputStream naming.resources.WARDirContext\$WARResource.streamContent()	19
[10268]String naming.SelectorContext.parseName(String)	11
[10269]Name naming.SelectorContext.parseName(Name)	11

Na Tabela C.4 são apresentados os grupos obtidos pela instância 12 da fase de agrupamento no sistema Tomcat com o respectivo resultado do cálculo de pontuação de cada uma das medidas de ordenação. A Coluna IT? indica se o grupo contém métodos referentes a instâncias de interesses transversais validadas, conforme Tabela A.3. A Coluna Métodos apresenta os identificadores dos métodos que compõem cada grupo, sendo que os métodos podem ser consultados na Tabela C.3.

Tabela C.4: Lista de grupos da instância 12 com pontuação: Tomcat

Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K1		6,0000	6,0000	3,0000	3,0000	[12]
K2		3,0000	3,0000	2,0000	2,0000	[26]
K3		14,0000	14,0000	9,0000	9,0000	[202]
K4		6,0000	6,0000	3,0000	3,0000	[227]
K5		5,0000	5,0000	3,0000	3,0000	[233]
K6		4,0000	4,0000	2,0000	2,0000	[386]
K7		3,0000	3,0000	2,0000	2,0000	[388]
K8		7,0000	7,0000	6,0000	6,0000	[392]
K9		4,0000	4,0000	2,0000	2,0000	[404]
K10		4,0000	4,0000	3,0000	3,0000	[563]
K11		4,0000	4,0000	2,0000	2,0000	[612]
K12		5,0000	5,0000	3,0000	3,0000	[623]
K13		4,0000	4,0000	2,0000	2,0000	[625]
K14		3,0000	3,0000	2,0000	2,0000	[627]
K15		5,0000	5,0000	3,0000	3,0000	[656]
K16		3,0000	3,0000	2,0000	2,0000	[814]
K17		3,0000	3,0000	2,0000	2,0000	[841]
K18		4,0000	4,0000	2,0000	2,0000	[859]
K19		4,0000	4,0000	2,0000	2,0000	[869]
K20		12,0000	12,0000	8,0000	8,0000	[922]
K21		3,0000	3,0000	2,0000	2,0000	[934]
K22		5,0000	5,0000	4,0000	4,0000	[943]
K23		4,0000	4,0000	3,0000	3,0000	[945]
K24		4,0000	4,0000	2,0000	2,0000	[1115]
K25		3,0000	3,0000	2,0000	2,0000	[1118]
K26		3,0000	3,0000	2,0000	2,0000	[1243]
K27		6,0000	6,0000	3,0000	3,0000	[1273]
K28		3,0000	3,0000	2,0000	2,0000	[1416]
K29		6,0000	6,0000	5,0000	5,0000	[1432]
K30		7,0000	7,0000	6,0000	6,0000	[1506]
K31		3,0000	3,0000	2,0000	2,0000	[1541]
K32		4,0000	4,0000	2,0000	2,0000	[1657]
K33		4,0000	4,0000	2,0000	2,0000	[1658]
K34		5,0000	5,0000	3,0000	3,0000	[1662]
K35		4,0000	4,0000	2,0000	2,0000	[1664]
K36		4,0000	4,0000	3,0000	3,0000	[1747]
K37		8,0000	8,0000	5,0000	5,0000	[1748]
K38		3,0000	3,0000	2,0000	2,0000	[1761]
K39		5,0000	5,0000	2,0000	2,0000	[1844]
K40		8,0000	8,0000	5,0000	5,0000	[1848]
K41		3,0000	3,0000	2,0000	2,0000	[1885]
K42		4,0000	4,0000	2,0000	2,0000	[1908]
K43		13,0000	13,0000	7,0000	7,0000	[1974]
K44		6,0000	6,0000	3,0000	3,0000	[1975]
K45		4,0000	4,0000	2,0000	2,0000	[1980]
K46		9,0000	9,0000	5,0000	5,0000	[2025]
K47		5,0000	5,0000	4,0000	4,0000	[2033]
K48		3,0000	3,0000	2,0000	2,0000	[2035]
K49		4,0000	4,0000	3,0000	3,0000	[2036]
K50		3,0000	3,0000	2,0000	2,0000	[2038]
K51		5,0000	5,0000	2,0000	2,0000	[2147]
K52	Sim	19,0000	19,0000	18,0000	18,0000	[2157]
K53		13,0000	13,0000	6,0000	6,0000	[2182]
K54		3,0000	3,0000	2,0000	2,0000	[2194]
K55		4,0000	4,0000	2,0000	2,0000	[2327]
K56		5,0000	5,0000	4,0000	4,0000	[2432]
K57		3,0000	3,0000	2,0000	2,0000	[2433]
K58		5,0000	5,0000	4,0000	4,0000	[2465]
K59		3,0000	3,0000	2,0000	2,0000	[2468]
K60		3,0000	3,0000	2,0000	2,0000	[2480]
K61		4,0000	4,0000	2,0000	2,0000	[2528]
K62		5,0000	5,0000	4,0000	4,0000	[2537]
K63		3,0000	3,0000	2,0000	2,0000	[2548]
K64		3,0000	3,0000	2,0000	2,0000	[2578]
K65		4,0000	4,0000	2,0000	2,0000	[2618]
K66	Sim	60,0000	60,0000	55,0000	55,0000	[2644]
K67		3,0000	3,0000	2,0000	2,0000	[2663]
K68		6,0000	6,0000	5,0000	5,0000	[2872]
K69		4,0000	4,0000	2,0000	2,0000	[2890]

Continua na próxima página ...

... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K70		5,0000	5,0000	4,0000	4,0000	[3070]
K71		4,0000	4,0000	3,0000	3,0000	[3071]
K72		5,0000	5,0000	2,0000	2,0000	[3180]
K73		4,0000	4,0000	2,0000	2,0000	[3366]
K74		7,0000	7,0000	4,0000	4,0000	[3573]
K75		7,0000	7,0000	3,0000	3,0000	[3603]
K76		4,0000	4,0000	3,0000	3,0000	[3774]
K77		3,0000	3,0000	2,0000	2,0000	[3904]
K78		3,0000	3,0000	2,0000	2,0000	[3930]
K79		6,0000	6,0000	5,0000	5,0000	[3946]
K80		4,0000	4,0000	2,0000	2,0000	[3948]
K81		5,0000	5,0000	2,0000	2,0000	[3956]
K82		4,0000	4,0000	2,0000	2,0000	[3959]
K83		4,0000	4,0000	2,0000	2,0000	[4020]
K84		14,0000	14,0000	10,0000	10,0000	[4022]
K85		3,0000	3,0000	2,0000	2,0000	[4034]
K86		3,0000	3,0000	2,0000	2,0000	[4042]
K87		5,0000	5,0000	4,0000	4,0000	[4044]
K88		4,0000	4,0000	3,0000	3,0000	[4046]
K89		4,0000	4,0000	2,0000	2,0000	[4198]
K90		3,0000	3,0000	2,0000	2,0000	[4201]
K91		6,0000	6,0000	5,0000	5,0000	[4228]
K92		10,0000	10,0000	8,0000	8,0000	[4355]
K93		3,0000	3,0000	2,0000	2,0000	[4424]
K94		3,0000	3,0000	2,0000	2,0000	[4431]
K95		3,0000	3,0000	2,0000	2,0000	[4437]
K96		4,0000	4,0000	2,0000	2,0000	[4460]
K97		3,0000	3,0000	2,0000	2,0000	[4558]
K98		4,0000	4,0000	3,0000	3,0000	[4576]
K99		3,0000	3,0000	2,0000	2,0000	[4694]
K100		5,0000	5,0000	4,0000	4,0000	[4853]
K101		4,0000	4,0000	3,0000	3,0000	[4907]
K102		6,0000	6,0000	3,0000	3,0000	[5045]
K103	Sim	57,0000	57,0000	46,0000	46,0000	[5046]
K104		4,0000	4,0000	3,0000	3,0000	[5097]
K105		3,0000	3,0000	2,0000	2,0000	[5127]
K106		3,0000	3,0000	2,0000	2,0000	[5148]
K107		5,0000	5,0000	4,0000	4,0000	[5149]
K108		4,0000	4,0000	2,0000	2,0000	[5152]
K109		4,0000	4,0000	3,0000	3,0000	[5172]
K110		3,0000	3,0000	2,0000	2,0000	[5188]
K111		4,0000	4,0000	3,0000	3,0000	[5191]
K112		4,0000	4,0000	3,0000	3,0000	[5225]
K113		4,0000	4,0000	3,0000	3,0000	[5226]
K114		9,0000	9,0000	4,0000	4,0000	[5298]
K115		5,0000	5,0000	4,0000	4,0000	[5364]
K116		3,0000	3,0000	2,0000	2,0000	[5416]
K117		5,0000	5,0000	4,0000	4,0000	[5472]
K118		6,0000	6,0000	5,0000	5,0000	[5474]
K119		7,0000	7,0000	3,0000	3,0000	[5578]
K120		4,0000	4,0000	3,0000	3,0000	[5657]
K121		5,0000	5,0000	2,0000	2,0000	[5672]
K122		3,0000	3,0000	2,0000	2,0000	[5677]
K123		3,0000	3,0000	2,0000	2,0000	[5682]
K124		4,0000	4,0000	3,0000	3,0000	[5688]
K125		5,0000	5,0000	3,0000	3,0000	[5710]
K126		4,0000	4,0000	3,0000	3,0000	[5744]
K127		3,0000	3,0000	2,0000	2,0000	[5745]
K128		3,0000	3,0000	2,0000	2,0000	[5752]
K129		5,0000	5,0000	2,0000	2,0000	[5772]
K130		9,0000	9,0000	5,0000	5,0000	[5828]
K131		4,0000	4,0000	3,0000	3,0000	[5848]
K132		6,0000	6,0000	5,0000	5,0000	[5905]
K133		3,0000	3,0000	2,0000	2,0000	[5975]
K134		3,0000	3,0000	2,0000	2,0000	[6038]
K135		13,0000	13,0000	6,0000	6,0000	[6107]
K136		3,0000	3,0000	2,0000	2,0000	[6114]
K137		8,0000	8,0000	4,0000	4,0000	[6177]
K138		3,0000	3,0000	2,0000	2,0000	[6208]
K139		5,0000	5,0000	2,0000	2,0000	[6209]
K140	Sim	15,0000	15,0000	8,0000	8,0000	[6228]
K141		22,0000	22,0000	14,0000	14,0000	[6229]
K142		15,0000	15,0000	11,0000	11,0000	[6267]
K143		3,0000	3,0000	2,0000	2,0000	[6395]
K144		4,0000	4,0000	2,0000	2,0000	[6405]
K145		5,0000	5,0000	3,0000	3,0000	[6440]
K146		4,0000	4,0000	2,0000	2,0000	[6459]
K147		4,0000	4,0000	2,0000	2,0000	[6516]
K148		11,0000	11,0000	6,0000	6,0000	[6526]
K149		8,0000	8,0000	4,0000	4,0000	[6529]
K150		8,0000	8,0000	7,0000	7,0000	[6536]
K151		15,0000	15,0000	9,0000	9,0000	[6547]
K152		5,0000	5,0000	2,0000	2,0000	[6548]
K153		12,0000	12,0000	9,0000	9,0000	[6550]
K154		4,0000	4,0000	3,0000	3,0000	[6556]
K155		5,0000	5,0000	3,0000	3,0000	[6559]
K156		6,0000	6,0000	3,0000	3,0000	[6560]
K157		6,0000	6,0000	4,0000	4,0000	[6585]
K158		4,0000	4,0000	2,0000	2,0000	[6667]
K159		5,0000	5,0000	3,0000	3,0000	[6699]
K160		3,0000	3,0000	2,0000	2,0000	[6701]

Continua na próxima página ...



... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K161		5,0000	5,0000	2,0000	2,0000	[6702]
K162		4,0000	4,0000	2,0000	2,0000	[6706]
K163		5,0000	5,0000	4,0000	4,0000	[6713]
K164		16,0000	16,0000	12,0000	12,0000	[6725]
K165		7,0000	7,0000	4,0000	4,0000	[6740]
K166		4,0000	4,0000	2,0000	2,0000	[6958]
K167		4,0000	4,0000	2,0000	2,0000	[6959]
K168		5,0000	5,0000	2,0000	2,0000	[7039]
K169		6,0000	6,0000	3,0000	3,0000	[7069]
K170		5,0000	5,0000	2,0000	2,0000	[7078]
K171		6,0000	6,0000	3,0000	3,0000	[7137]
K172		5,0000	5,0000	2,0000	2,0000	[7153]
K173		4,0000	4,0000	2,0000	2,0000	[7340]
K174		4,0000	4,0000	2,0000	2,0000	[7385]
K175		6,0000	6,0000	3,0000	3,0000	[7401]
K176		4,0000	4,0000	2,0000	2,0000	[7403]
K177	Sim	94,0000	94,0000	68,0000	68,0000	[7408]
K178		10,0000	10,0000	6,0000	6,0000	[7423]
K179		4,0000	4,0000	3,0000	3,0000	[7457]
K180		3,0000	3,0000	2,0000	2,0000	[7471]
K181	Sim	7,0000	7,0000	3,0000	3,0000	[7475]
K182		7,0000	7,0000	3,0000	3,0000	[7476]
K183		4,0000	4,0000	2,0000	2,0000	[7479]
K184		3,0000	3,0000	2,0000	2,0000	[7484]
K185		5,0000	5,0000	2,0000	2,0000	[7506]
K186		5,0000	5,0000	4,0000	4,0000	[7537]
K187		3,0000	3,0000	2,0000	2,0000	[7557]
K188		9,0000	9,0000	4,0000	4,0000	[7561]
K189		3,0000	3,0000	2,0000	2,0000	[7569]
K190		3,0000	3,0000	2,0000	2,0000	[7600]
K191		4,0000	4,0000	3,0000	3,0000	[7642]
K192		3,0000	3,0000	2,0000	2,0000	[7738]
K193		5,0000	5,0000	2,0000	2,0000	[7739]
K194		4,0000	4,0000	2,0000	2,0000	[7806]
K195		5,0000	5,0000	2,0000	2,0000	[7820]
K196		4,0000	4,0000	3,0000	3,0000	[7835]
K197	Sim	13,0000	13,0000	12,0000	12,0000	[7871]
K198		3,0000	3,0000	2,0000	2,0000	[7893]
K199		10,0000	10,0000	9,0000	9,0000	[7909]
K200		5,0000	5,0000	3,0000	3,0000	[7937]
K201		8,0000	8,0000	4,0000	4,0000	[7938]
K202		4,0000	4,0000	2,0000	2,0000	[7950]
K203	Sim	16,0000	16,0000	15,0000	15,0000	[7993]
K204		6,0000	6,0000	3,0000	3,0000	[7995]
K205		21,0000	21,0000	18,0000	18,0000	[8003]
K206		4,0000	4,0000	2,0000	2,0000	[8004]
K207		5,0000	5,0000	2,0000	2,0000	[8061]
K208		3,0000	3,0000	2,0000	2,0000	[8062]
K209		5,0000	5,0000	3,0000	3,0000	[8072]
K210		3,0000	3,0000	2,0000	2,0000	[8053]
K211		5,0000	5,0000	2,0000	2,0000	[8081]
K212		4,0000	4,0000	2,0000	2,0000	[8100]
K213		3,0000	3,0000	2,0000	2,0000	[8112]
K214		6,0000	6,0000	3,0000	3,0000	[8114]
K215		6,0000	6,0000	5,0000	5,0000	[8132]
K216		3,0000	3,0000	2,0000	2,0000	[8150]
K217		4,0000	4,0000	2,0000	2,0000	[8163]
K218		45,0000	45,0000	42,0000	42,0000	[8166]
K219		15,0000	15,0000	12,0000	12,0000	[8167]
K220		3,0000	3,0000	2,0000	2,0000	[8177]
K221		3,0000	3,0000	2,0000	2,0000	[8185]
K222		5,0000	5,0000	2,0000	2,0000	[8187]
K223	Sim	22,0000	22,0000	18,0000	18,0000	[8213]
K224		4,0000	4,0000	2,0000	2,0000	[8215]
K225		14,0000	14,0000	8,0000	8,0000	[8220]
K226		4,0000	4,0000	2,0000	2,0000	[8221]
K227		5,0000	5,0000	2,0000	2,0000	[8225]
K228		4,0000	4,0000	2,0000	2,0000	[8251]
K229		6,0000	6,0000	5,0000	5,0000	[8306]
K230		15,0000	15,0000	9,0000	9,0000	[8309]
K231		5,0000	5,0000	2,0000	2,0000	[8423]
K232		3,0000	3,0000	2,0000	2,0000	[8476]
K233		5,0000	5,0000	2,0000	2,0000	[8503]
K234		16,0000	16,0000	12,0000	12,0000	[8513]
K235		6,0000	6,0000	3,0000	3,0000	[8516]
K236		8,0000	8,0000	5,0000	5,0000	[8521]
K237	Sim	46,0000	46,0000	26,0000	26,0000	[8523]
K238		27,0000	27,0000	18,0000	18,0000	[8524]
K239	Sim	23,0000	23,0000	22,0000	22,0000	[8604]
K240		4,0000	4,0000	2,0000	2,0000	[8617]
K241		9,0000	9,0000	4,0000	4,0000	[8621]
K242		5,0000	5,0000	3,0000	3,0000	[8627]
K243		5,0000	5,0000	2,0000	2,0000	[8646]
K244		7,0000	7,0000	4,0000	4,0000	[8684]
K245	Sim	20,0000	20,0000	19,0000	19,0000	[8689]
K246		7,0000	7,0000	6,0000	6,0000	[8695]
K247		3,0000	3,0000	2,0000	2,0000	[8696]
K248		4,0000	4,0000	2,0000	2,0000	[8766]
K249		4,0000	4,0000	2,0000	2,0000	[8773]
K250		7,0000	7,0000	4,0000	4,0000	[8779]
K251		4,0000	4,0000	2,0000	2,0000	[8782]

Continua na próxima página ...

... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIrank	Métodos
K252		5,0000	5,0000	3,0000	3,0000	[8807]
K253		4,0000	4,0000	2,0000	2,0000	[8811]
K254		5,0000	5,0000	2,0000	2,0000	[8877]
K255		4,0000	4,0000	2,0000	2,0000	[8882]
K256		5,0000	5,0000	3,0000	3,0000	[8943]
K257		4,0000	4,0000	2,0000	2,0000	[8949]
K258		3,0000	3,0000	2,0000	2,0000	[8953]
K259		5,0000	5,0000	4,0000	4,0000	[9122]
K260		6,0000	6,0000	3,0000	3,0000	[9131]
K261		13,0000	13,0000	9,0000	9,0000	[9146]
K262		4,0000	4,0000	3,0000	3,0000	[9171]
K263		5,0000	5,0000	4,0000	4,0000	[9174]
K264		3,0000	3,0000	2,0000	2,0000	[9204]
K265		3,0000	3,0000	2,0000	2,0000	[9210]
K266		10,0000	10,0000	9,0000	9,0000	[9218]
K267		3,0000	3,0000	2,0000	2,0000	[9256]
K268		3,0000	3,0000	2,0000	2,0000	[9282]
K269		4,0000	4,0000	3,0000	3,0000	[9285]
K270		4,0000	4,0000	2,0000	2,0000	[9315]
K271		4,0000	4,0000	2,0000	2,0000	[9344]
K272		3,0000	3,0000	2,0000	2,0000	[9364]
K273	Sim	21,0000	21,0000	17,0000	17,0000	[9463]
K274		4,0000	4,0000	2,0000	2,0000	[9511]
K275		4,0000	4,0000	2,0000	2,0000	[9518]
K276		3,0000	3,0000	2,0000	2,0000	[9579]
K277	Sim	26,0000	26,0000	25,0000	25,0000	[9590]
K278		11,0000	11,0000	6,0000	6,0000	[9634]
K279		3,0000	3,0000	2,0000	2,0000	[9636]
K280		3,0000	3,0000	2,0000	2,0000	[9647]
K281		3,0000	3,0000	2,0000	2,0000	[9648]
K282		3,0000	3,0000	2,0000	2,0000	[9668]
K283		4,0000	4,0000	2,0000	2,0000	[9698]
K284		3,0000	3,0000	2,0000	2,0000	[9747]
K285		10,0000	10,0000	8,0000	8,0000	[9772]
K286		4,0000	4,0000	3,0000	3,0000	[9832]
K287		5,0000	5,0000	4,0000	4,0000	[9859]
K288		3,0000	3,0000	2,0000	2,0000	[9869]
K289		6,0000	6,0000	5,0000	5,0000	[9978]
K290		3,0000	3,0000	2,0000	2,0000	[10010]
K291	Sim	26,0000	26,0000	25,0000	25,0000	[10118]
K292	Sim	25,0000	25,0000	24,0000	24,0000	[10119]
K293		6,0000	6,0000	5,0000	5,0000	[10120]
K294	Sim	24,0000	24,0000	23,0000	23,0000	[10124]
K295		4,0000	4,0000	2,0000	2,0000	[10188]
K296		3,0000	3,0000	2,0000	2,0000	[10191]
K297		10,0000	7,4667	8,0000	5,9733	[9659] [9661]
K298		7,0000	5,1100	6,0000	4,3800	[4620] [4610] [4617]
K299		30,0000	22,8114	57,0000	43,3416	[10135] [10038] [10235]
K300	Sim	101,0000	70,0358	244,0000	169,1953	[8531] [8834] [8941]
K301		45,0000	28,8392	87,0000	55,7558	[4525] [5305] [5373]
K302	Sim	30,0000	19,0399	40,0000	25,3865	[2266] [2622]
K303		4,0000	2,5352	4,0000	2,5352	[8734] [8735]
K304		5,0000	3,1655	6,0000	3,7986	[7314] [7451]
K305	Sim	38,0000	23,9985	48,0000	30,3138	[2265] [2621]
K306		4,0000	2,4971	4,0000	2,4971	[5123] [5177]
K307		24,0000	14,8917	32,0000	19,8557	[4526] [5306]
K308		12,0000	7,3800	8,0000	4,9200	[5037] [5038]
K309	Sim	21,0000	13,1907	39,0000	24,4971	[2269] [2625] [3035]
K310		9,0000	5,4701	5,0000	3,0390	[4274] [4275]
K311		10,0000	6,0667	6,0000	3,6400	[1883] [1884]
K312	Sim	45,0000	30,7301	460,0000	314,1299	[7578] [7765] [3380] [3127] [3160] [3204] [7719] [1444] [4240] [981] [4082] [7616] [7755] [7729] [7636] [7708] [7597] [7675] [313] [384] [7737] [7723] [7726]
K313		24,0000	14,4302	22,0000	13,2277	[4520] [5300]
K314		23,0000	14,8937	39,0000	25,2545	[4528] [5307] [5376]
K315		10,0000	5,9643	6,0000	3,5786	[1494] [1495]
K316	Sim	38,0000	23,8075	165,0000	103,3747	[8995] [7997] [8858] [8530] [8935]
K317		13,0000	7,7291	18,0000	10,7018	[1240] [1321]
K318	Sim	33,0000	27,6381	286,0000	239,5301	[11] [10] [284] [288] [264] [280] [276] [292] [55] [260] [250] [246] [258] [274] [268] [272] [47] [83]
K319		4,0000	2,3486	4,0000	2,3486	[4706] [4711]
K320		13,0000	7,5714	16,0000	9,3187	[2636] [4753]
K321		10,0000	5,7643	11,0000	6,3407	[1821] [1829]
K322		17,0000	9,7705	18,0000	10,3453	[7194] [7280]
K323		18,0000	10,3371	24,0000	13,7829	[6838] [6964]
K324		23,0000	13,1235	17,0000	9,7000	[28] [30]
K325		13,0000	7,4100	16,0000	9,1200	[415] [1530]
K326		13,0000	7,4100	16,0000	9,1200	[3633] [4306]
K327		10,0000	5,6667	7,0000	3,9667	[5881] [5882]
K328		3,0000	1,8656	10,0000	6,2186	[8698] [8704] [8705] [8697] [8703]
K329	Sim	21,0000	14,2357	66,0000	44,7409	[2977] [3320] [3144] [3189] [2262] [2617]
K330		16,0000	10,7930	56,0000	37,7755	[6319] [6278] [6392] [6419] [6345] [6413] [6428]
K331		7,0000	4,9167	9,0000	6,3214	[9917] [10013] [10198]
K332		28,0000	19,2869	168,0000	115,7213	[6071] [6085] [6097] [6080] [6074] [6077] [6190] [6052] [6057] [6241] [6237] [6239]
K333		5,0000	2,7967	4,0000	2,2373	[7033] [7040]
K334		5,0000	2,8299	6,0000	3,3958	[5921] [5917] [5919]
K335		20,0000	11,0000	12,0000	6,6000	[7480] [7481]
K336		14,0000	9,0834	35,0000	22,7084	[6836] [6911] [7028] [6967] [7046]
K337		5,0000	3,1297	6,0000	3,7557	[1265] [1141] [1301]
K338		12,0000	7,3643	25,0000	15,3422	[1319] [1262] [1345] [1139] [1299]

Continua na próxima página ...

... continuação da página anterior									
Grupo	IT?	GSRank	GSIRank	GFRank	GFIrank	Métodos			
K339		8,0000	5,0791	6,0000	3,8093	[7393]	[7391]	[7392]	
K340		9,0000	4,8413	10,0000	5,3792	[489]	[881]		
K341		9,0000	4,8413	10,0000	5,3792	[3697]	[3980]		
K342		9,0000	4,8327	8,0000	4,2958	[5221]	[6027]		
K343		10,0000	5,3627	7,0000	3,7539	[2177]	[2179]		
K344		8,0000	4,2708	8,0000	4,2708	[5629]	[5685]		
K345		14,0000	7,4317	14,0000	7,4317	[4969]	[4970]		
K346		15,0000	8,8533	42,0000	24,7892	[1114]	[4197]	[4524]	[5303] [5405] [811] [3901]
K347		5,0000	3,1003	6,0000	3,7204	[5461]	[1025]	[4126]	
K348		43,0000	24,4860	103,0000	58,6525	[8778]	[8771]	[8774]	
K349		6,0000	3,1500	6,0000	3,1500	[9788]	[9797]		
K350		7,0000	4,6833	12,0000	8,0285	[2652]	[3152]	[3092]	[3200]
K351		4,0000	2,0819	4,0000	2,0819	[9925]	[10015]		
K352		121,0000	68,3828	122,0000	68,9479	[7975]	[7984]	[7982]	[7976] [7979]
K353		11,0000	5,7031	10,0000	5,1846	[1817]	[1819]		
K354		8,0000	4,1440	8,0000	4,1440	[3286]	[5073]		
K355	Sim	22,0000	11,3117	21,0000	10,7975	[2463]	[2464]		
K356		5,0000	2,5667	4,0000	2,0533	[4393]	[4395]		
K357		10,0000	5,1333	10,0000	5,1333	[7193]	[7279]		
K358		15,0000	9,4342	43,0000	27,0448	[6362]	[6429]	[6393]	[6414] [6279] [6326] [6346]
K359		3,0000	1,8104	6,0000	3,6208	[9690]	[9688]	[9689]	
K360		3,0000	1,5364	4,0000	2,0486	[4476]	[4477]		
K361		5,0000	2,5607	8,0000	4,0971	[4775]	[4776]		
K362		7,0000	3,8595	9,0000	4,9622	[5014]	[4960]	[4997]	
K363		11,0000	6,2233	12,0000	6,7891	[6028]	[5222]	[5256]	
K364		13,0000	6,9003	16,0000	8,4927	[6029]	[5223]	[5257]	
K365		5,0000	3,1281	6,0000	3,7537	[5451]	[1033]	[4134]	
K366		6,0000	3,0231	8,0000	4,0308	[4704]	[4714]		
K367		10,0000	5,0133	10,0000	5,0133	[2349]	[3001]		
K368		7,0000	3,5093	6,0000	3,0080	[2372]	[3029]		
K369		3,0000	1,6682	10,0000	5,5606	[8012]	[8013]	[8015]	[8011] [8014]
K370		10,0000	4,9569	8,0000	3,9655	[8909]	[8930]		
K371		7,0000	3,4689	6,0000	2,9733	[15]	[27]		
K372		5,0000	2,4721	4,0000	1,9776	[7509]	[7511]		
K373		8,0000	3,9429	13,0000	6,4071	[5262]	[5263]		
K374		4,0000	1,9600	4,0000	1,9600	[335]	[336]		
K375		13,0000	8,4602	32,0000	20,8251	[5556]	[5580]	[5584]	[5588] [5559] [5570] [5565] [5574]
K376		8,0000	4,6027	12,0000	6,9040	[461]	[1195]	[1365]	[1559]
K377		5,0000	2,4359	4,0000	1,9487	[2641]	[4755]		
K378		9,0000	4,7355	20,0000	10,5232	[6817]	[6818]	[3977]	[3978]
K379		7,0000	3,3982	7,0000	3,3982	[8798]	[8800]		
K380		11,0000	5,3265	7,0000	3,3896	[2173]	[2174]		
K381		7,0000	3,3890	4,0000	1,9366	[1509]	[1510]		
K382		10,0000	4,8415	6,0000	2,9049	[4291]	[4292]		
K383		4,0000	1,9357	4,0000	1,9357	[6081]	[6191]		
K384		9,0000	4,3551	8,0000	3,8712	[3243]	[5057]		
K385		10,0000	4,8333	5,0000	2,4167	[5254]	[5258]		
K386		4,0000	1,9286	4,0000	1,9286	[6722]	[6724]		
K387		5,0000	2,4079	6,0000	2,8895	[8818]	[8827]		
K388		8,0000	3,8523	8,0000	3,8523	[2263]	[2619]		
K389		5,0000	2,4013	4,0000	1,9210	[1263]	[1328]		
K390		10,0000	4,8000	4,0000	1,9200	[1053]	[4150]		
K391		11,0000	6,5592	17,0000	10,1370	[6280]	[6359]	[6350]	[6398]
K392		8,0000	3,8400	8,0000	3,8400	[7195]	[7281]		
K393	Sim	23,0000	11,0400	22,0000	10,5600	[10268]	[10269]		
K394		9,0000	4,6513	9,0000	4,6513	[6032]	[5224]	[5259]	
K395		4,0000	1,8926	4,0000	1,8926	[6430]	[6431]		
K396		4,0000	1,8910	4,0000	1,8910	[3206]	[3208]		
K397		11,0000	5,2947	8,0000	3,8507	[5473]	[944]	[4045]	
K398		6,0000	2,8179	6,0000	2,8179	[6287]	[6711]		
K399		5,0000	2,3436	4,0000	1,8749	[3541]	[3553]		
K400		8,0000	3,7440	6,0000	2,8080	[2370]	[3028]		
K401		8,0000	3,7440	6,0000	2,8080	[3277]	[5071]		
K402		8,0000	3,7440	6,0000	2,8080	[3278]	[5072]		
K403		5,0000	2,3286	4,0000	1,8629	[7200]	[7287]		
K404		8,0000	4,1749	8,0000	4,1749	[602]	[1202]	[1566]	
K405		10,0000	5,3620	18,0000	9,6517	[1138]	[1298]	[1318]	[1261] [1344]
K406		8,0000	3,6800	6,0000	2,7600	[382]	[5546]		
K407		9,0000	4,1400	7,0000	3,2200	[4847]	[4908]		
K408		6,0000	3,3162	6,0000	3,3162	[6512]	[6513]	[6515]	
K409		7,0000	3,2132	6,0000	2,7542	[6072]	[6189]		
K410		5,0000	2,2874	4,0000	1,8299	[690]	[730]		
K411		5,0000	2,2874	4,0000	1,8299	[6617]	[6683]		
K412		5,0000	2,2686	4,0000	1,8149	[2340]	[2991]		
K413		5,0000	2,2686	4,0000	1,8149	[3341]	[7809]		
K414		3,0000	1,3733	6,0000	2,7467	[1875]	[1871]	[1873]	
K415		6,0000	2,7153	4,0000	1,8102	[6778]	[6799]		
K416		38,0000	21,7994	68,0000	39,0094	[8305]	[8398]	[8438]	[8356] [8377] [8385] [8369] [8374] [8347] [8344]
						[8380]	[8403]	[8359]	[8364] [8496] [8420] [8483] [8390] [8393] [8308]
						[8320]	[8486]	[8330]	[8323] [8333] [8328] [8338] [8341] [8349] [8414]
						[8409]	[8499]	[8362]	[8479]
K417		5,0000	2,2571	4,0000	1,8057	[3258]	[5059]		
K418		10,0000	5,4380	16,0000	8,7009	[6366]	[6355]	[6424]	[6285] [6320]
K419		9,0000	4,0930	12,0000	5,4573	[5448]	[5538]	[916]	[4015]
K420		4,0000	1,7800	4,0000	1,7800	[6061]	[6124]		
K421		7,0000	3,1044	6,0000	2,6610	[3239]	[5054]		
K422		5,0000	2,2104	4,0000	1,7683	[486]	[1385]		
K423		7,0000	3,0883	4,0000	1,7648	[2975]	[3337]		
K424		9,0000	4,0336	9,0000	4,0336	[9067]	[9019]	[9183]	
K425		13,0000	6,2922	13,0000	6,2922	[10063]	[10073]	[10117]	
K426		4,0000	1,7533	4,0000	1,7533	[7965]	[7966]		

Continua na próxima página ...

... continuação da página anterior							
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos	
K427		4,0000	1,7533	4,0000	1,7533	[7970]	[7971]
K428		3,0000	1,3470	6,0000	2,6940	[515]	[516] [517]
K429		3,0000	1,3470	6,0000	2,6940	[3734]	[3735] [3736]
K430		18,0000	8,9798	30,0000	14,9663	[2264]	[2620] [2271] [2627]
K431		5,0000	2,1727	4,0000	1,7382	[5192]	[5193]
K432		12,0000	5,2086	14,0000	6,0767	[7568]	[7570]
K433		20,0000	8,6737	14,0000	6,0716	[6192]	[6193]
K434		4,0000	1,7333	4,0000	1,7333	[9889]	[9895]
K435		6,0000	2,5892	6,0000	2,5892	[2270]	[2626]
K436		6,0000	2,5800	4,0000	1,7200	[7192]	[7278]
K437		6,0000	2,5714	4,0000	1,7143	[3536]	[3559]
K438		21,0000	8,9783	17,0000	7,2681	[907]	[908]
K439		8,0000	3,8747	12,0000	5,8121	[6719]	[6773] [6794]
K440		4,0000	1,7040	4,0000	1,7040	[439]	[504]
K441		4,0000	1,7040	4,0000	1,7040	[3654]	[3723]
K442		6,0000	2,5560	6,0000	2,5560	[9753]	[9763]
K443		5,0000	2,1262	4,0000	1,7010	[561]	[562]
K444		5,0000	2,1262	4,0000	1,7010	[3772]	[3773]
K445		6,0000	2,5500	4,0000	1,7000	[1113]	[4196]
K446		6,0000	2,5457	4,0000	1,6971	[8244]	[8246]
K447		6,0000	2,5400	4,0000	1,6933	[5220]	[6030]
K448		6,0000	3,1359	8,0000	4,1812	[6321]	[6286] [6356] [6425]
K449		4,0000	1,6897	6,0000	2,5345	[9236]	[9237]
K450		10,0000	5,4168	16,0000	8,6668	[4530]	[3822] [5408] [1112] [4195]
K451		40,0000	30,1404	134,0000	100,9703	[4359]	[4416] [1515] [2629] [3215] [3253] [3282] [378] [1745] [4297] [4464] [5862] [5411] [823] [3912] [4964] [5504] [322] [5383] [7684] [7648] [7608] [7710] [7750] [992] [4093]
K452		14,0000	5,8646	11,0000	4,6079	[5299]	[5311]
K453		6,0000	2,5080	4,0000	1,6720	[2346]	[2998]
K454		6,0000	2,5080	4,0000	1,6720	[2348]	[3000]
K455		6,0000	2,5080	4,0000	1,6720	[2355]	[3007]
K456		6,0000	2,5080	4,0000	1,6720	[2356]	[3008]
K457		6,0000	2,5080	4,0000	1,6720	[2360]	[3017]
K458		6,0000	2,5080	4,0000	1,6720	[2363]	[3020]
K459		6,0000	2,5080	4,0000	1,6720	[2369]	[3026]
K460		6,0000	2,5080	4,0000	1,6720	[2375]	[3032]
K461		6,0000	2,5080	4,0000	1,6720	[2387]	[3047]
K462		4,0000	1,6720	4,0000	1,6720	[3327]	[7797]
K463		6,0000	2,5080	4,0000	1,6720	[3330]	[7800]
K464		6,0000	2,8229	6,0000	2,8229	[5070]	[3274] [5846]
K465		3,0000	1,2518	4,0000	1,6691	[5043]	[5044]
K466		6,0000	2,4882	4,0000	1,6588	[6973]	[7017]
K467		20,0000	8,5047	21,0000	8,9299	[3543]	[3550] [3551]
K468		7,0000	2,9121	6,0000	2,4961	[3625]	[7209] [7244]
K469		7,0000	2,9121	6,0000	2,4961	[3627]	[7211] [7246]
K470		6,0000	3,2757	7,0000	3,8217	[4695]	[4699] [4700]
K471		8,0000	3,9546	10,0000	4,9433	[1300]	[1140] [1346] [1264] [1320]
K472		5,0000	2,0542	5,0000	2,0542	[6898]	[6899]
K473		6,0000	2,4600	4,0000	1,6400	[3190]	[4337]
K474		7,0000	2,8910	6,0000	2,4780	[7177]	[7230] [7256]
K475		7,0000	2,8910	6,0000	2,4780	[7178]	[7231] [7257]
K476		7,0000	2,8910	6,0000	2,4780	[7181]	[7234] [7260]
K477		7,0000	2,8910	6,0000	2,4780	[7183]	[7236] [7262]
K478		10,0000	4,0818	15,0000	6,1227	[9720]	[9729]
K479		8,0000	3,7464	12,0000	5,6196	[3325]	[3326] [7795] [7796]
K480		6,0000	2,4480	4,0000	1,6320	[5609]	[5643]
K481		5,0000	2,0381	4,0000	1,6305	[8874]	[8875]
K482		6,0000	2,4400	4,0000	1,6267	[3973]	[6813]
K483		6,0000	2,4400	4,0000	1,6267	[3975]	[6815]
K484		7,0000	2,8389	7,0000	2,8389	[8821]	[8822]
K485		4,0000	1,6940	9,0000	3,8116	[1421]	[1419] [1420]
K486		14,0000	8,0358	38,0000	21,8114	[6678]	[6348] [6363] [6324] [6432] [6626] [6716]
K487		15,0000	8,3921	40,0000	22,3789	[6679]	[6349] [6364] [6325] [6433] [6627] [6717]
K488		6,0000	2,4171	4,0000	1,6114	[1143]	[1303]
K489		13,0000	5,7010	16,0000	7,0166	[5447]	[5537] [915] [4014]
K490		7,0000	2,8125	6,0000	2,4107	[4010]	[4011]
K491		7,0000	3,0094	8,0000	3,4393	[3718]	[4112] [3665] [3847]
K492		8,0000	3,2000	5,0000	2,0000	[3949]	[3954]
K493		6,0000	2,4000	6,0000	2,4000	[5866]	[5867]
K494		7,0000	2,7903	6,0000	2,3917	[911]	[912]
K495		35,0000	17,3168	78,0000	38,5916	[5534]	[5441] [1024] [4125] [910] [4009]
K496		8,0000	3,1822	4,0000	1,5911	[7165]	[7166]
K497		5,0000	2,2524	6,0000	2,7029	[10214]	[9955] [10025]
K498		5,0000	2,2413	6,0000	2,6895	[10216]	[9959] [10027]
K499		7,0000	2,8359	6,0000	2,4308	[4672]	[4670] [4671]
K500		6,0000	3,0746	14,0000	7,1742	[9728]	[9727] [9721] [9722]
K501		8,0000	3,6705	11,0000	5,0469	[8477]	[8455] [8453] [8454]
K502		5,0000	1,9593	5,0000	1,9593	[397]	[398]
K503		8,0000	3,1346	6,0000	2,3510	[9167]	[9169]
K504		6,0000	2,3481	6,0000	2,3481	[4702]	[4712]
K505		15,0000	7,0203	27,0000	12,6366	[5007]	[4925] [4952] [4953] [4773] [4897] [4818] [4845]
K506	Sim	22,0000	10,1144	30,0000	13,7924	[7868]	[7867] [7870]
K507		7,0000	2,7873	6,0000	2,3891	[9061]	[9037] [9200]
K508		5,0000	1,9429	4,0000	1,5543	[5563]	[5567]
K509		20,0000	9,9376	16,0000	7,9501	[3560]	[3562] [3537] [3539]
K510		5,0000	1,9381	4,0000	1,5505	[1621]	[1638]
K511		22,0000	8,4700	14,0000	5,3900	[2183]	[2184]
K512		5,0000	1,9155	4,0000	1,5324	[3219]	[3220]
K513		26,0000	11,0186	28,0000	11,8662	[5439]	[4007] [5533]
K514	Sim	74,0000	28,2666	103,0000	39,3441	[4603]	[4623]
K515		10,0000	4,6527	7,0000	3,2569	[6558]	[6744] [6747]

Continua na próxima página ...

... continuação da página anterior								
Grupo	IT?	GSRank	GSIRank	GFRank	GFI Rank	Métodos		
K516		6,0000	2,2895	4,0000	1,5263	[8491]	[8502]	
K517		7,0000	2,6678	5,0000	1,9056	[3588]	[3605]	
K518	Sim	14,0000	5,3338	13,0000	4,9528	[8722]	[8724]	
K519		6,0000	2,8392	28,0000	13,2495	[8730]	[8711]	[8733] [8706] [8701] [8702] [8732] [8731] [8737]
K520		22,0000	9,9899	14,0000	6,3572	[6530]	[6535]	[6531] [6532] [6534]
K521		23,0000	10,6593	60,0000	27,8068	[6835]	[6963]	[7032] [7041] [6895] [6909] [7004] [7026]
K522	Sim	92,0000	54,5295	1125,0000	666,8004	[3150]	[3365]	[3704] [2632] [3065] [5414] [1122] [3831] [4207] [5757] [3285] [5865] [3218] [3256] [1526] [4303] [1752] [5510] [7687] [325] [381] [7753] [995] [4096] [7651] [7611] [7715] [4469] [5386] [830] [3919] [409] [4419] [5265] [4362] [4967] [4917] [4857] [4933] [5002] [4786] [4825]
K523		5,0000	1,8950	4,0000	1,5160	[4377]	[4399]	
K524		6,0000	2,2650	5,0000	1,8875	[8605]	[8610]	
K525	Sim	68,0000	28,3939	132,0000	55,1176	[5436]	[5530]	[905] [4004]
K526		6,0000	2,2638	4,0000	1,5092	[936]	[947]	
K527		6,0000	2,2832	6,0000	2,2832	[1103]	[451]	[806]
K528		10,0000	3,7643	11,0000	4,1407	[9371]	[9379]	
K529		7,0000	2,6246	9,0000	3,3745	[7959]	[7960]	
K530	Sim	83,0000	48,2153	1009,0000	586,1358	[3195]	[3149]	[3364] [3703] [2631] [3062] [5413] [3829] [1121] [4206] [825] [3914] [1751] [324] [994] [4095] [5509] [7686] [7610] [7714] [7650] [380] [7752] [1518] [4300] [4418] [5756] [4468] [5864] [5385] [3217] [3255] [3284] [408] [5264] [4361] [4966] [4932] [5001] [4785] [4824] [4856] [4916]
K531		7,0000	2,6679	8,0000	3,0490	[4186]	[3817]	[3677] [3896]
K532		5,0000	1,8672	7,0000	2,6140	[5922]	[5923]	
K533		9,0000	3,3464	11,0000	4,0900	[5909]	[5910]	
K534		12,0000	6,0657	18,0000	9,0985	[3148]	[2635]	[3066] [3199]
K535		6,0000	2,7112	8,0000	3,6149	[10215]	[9957]	[10026]
K536		10,0000	4,2324	11,0000	4,6556	[319]	[314]	[320]
K537		10,0000	4,1680	13,0000	5,4184	[6151]	[6147]	[6179]
K538		5,0000	1,8464	4,0000	1,4771	[8992]	[8993]	
K539		4,0000	1,5689	7,0000	2,7456	[2876]	[2879]	[2881]
K540		6,0000	2,2084	4,0000	1,4723	[4036]	[4048]	
K541		11,0000	4,0470	9,0000	3,3112	[4643]	[4644]	
K542	Sim	47,0000	35,8543	196,0000	149,5199	[4358]	[4415]	[2628] [3252] [3214] [3281] [377] [1744] [5410] [822] [3911] [1514] [4296] [4463] [5861] [4963] [5503] [321] [5382] [7683] [7647] [7607] [7709] [7749] [991] [4092]
K543		9,0000	3,7504	8,0000	3,3337	[9862]	[9836]	[9860]
K544		8,0000	3,3603	9,0000	3,7803	[621]	[622]	[619] [620]
K545		22,0000	11,4063	38,0000	19,7018	[5452]	[1026]	[4127] [5453] [1027] [4128]
K546		5,0000	1,9598	7,0000	2,7437	[7448]	[7447]	[7450]
K547		5,0000	1,7964	4,0000	1,4371	[7968]	[7973]	
K548		14,0000	5,7456	15,0000	6,1560	[3542]	[3533]	[3556]
K549		6,0000	2,1450	7,0000	2,5025	[10225]	[10226]	
K550		9,0000	3,2169	5,0000	1,7872	[8324]	[8334]	
K551		4,0000	2,0559	29,0000	14,9050	[2861]	[2860]	[2865] [2863] [2864] [2862] [2866] [2868] [2867] [2870] [2869] [2871]
K552		7,0000	2,4967	4,0000	1,4267	[3585]	[3606]	
K553		5,0000	2,2574	6,0000	2,7089	[7958]	[7955]	[7956]
K554		7,0000	2,4960	5,0000	1,7828	[2158]	[2159]	
K555		9,0000	3,6164	8,0000	3,2146	[465]	[603]	[1176]
K556		14,0000	6,2350	27,0000	12,0247	[815]	[4522]	[5304] [3824] [3905]
K557		41,0000	17,4846	99,0000	42,2188	[8772]	[8775]	[8768] [8769]
K558		7,0000	2,4672	5,0000	1,7623	[8212]	[8223]	
K559		10,0000	4,5036	14,0000	6,3051	[8824]	[8825]	[8826]
K560		8,0000	2,7911	7,0000	2,4422	[2426]	[2427]	
K561		4,0000	1,3867	4,0000	1,3867	[6415]	[6416]	
K562		26,0000	10,9633	29,0000	12,2283	[5438]	[4006]	[5531]
K563		15,0000	5,1971	10,0000	3,4647	[2024]	[2027]	
K564	Sim	22,0000	7,6183	21,0000	7,2720	[2203]	[2213]	
K565		5,0000	1,7285	4,0000	1,3828	[5971]	[5972]	
K566		10,0000	4,7397	14,0000	6,6356	[4304]	[4305]	[3632] [3634]
K567		10,0000	4,2851	14,0000	5,9992	[414]	[1528]	[416] [1527]
K568		5,0000	1,6964	4,0000	1,3571	[7851]	[7853]	
K569		6,0000	2,0271	5,0000	1,6893	[8740]	[8741]	
K570		10,0000	3,7850	18,0000	6,8129	[9353]	[9358]	[9354] [9362]
K571		6,0000	2,4141	7,0000	2,8165	[5197]	[5195]	[5196]
K572		7,0000	2,9448	10,0000	4,2068	[3340]	[7808]	[3317] [7790]
K573		5,0000	1,6714	6,0000	2,0057	[7643]	[7645]	
K574		9,0000	3,4008	12,0000	4,5344	[3147]	[3094]	[2633] [3198]
K575	Sim	20,0000	7,2843	19,0000	6,9200	[8855]	[8854]	[8856]
K576		7,0000	2,3242	5,0000	1,6601	[5879]	[5880]	
K577		4,0000	1,3251	4,0000	1,3251	[2482]	[2484]	
K578		4,0000	1,3200	5,0000	1,6500	[5717]	[5718]	
K579		8,0000	2,6351	5,0000	1,6469	[8456]	[8469]	
K580		10,0000	4,0362	12,0000	4,8434	[1101]	[448]	[790]
K581		11,0000	4,2380	14,0000	5,3938	[3674]	[3880]	[3815] [4184]
K582		10,0000	3,2758	7,0000	2,2930	[1816]	[1887]	
K583		17,0000	7,8280	17,0000	7,8280	[5661]	[5664]	[5665]
K584		10,0000	3,2507	10,0000	3,2507	[5190]	[5202]	
K585		11,0000	4,4896	14,0000	5,7140	[7191]	[7277]	[7190] [7276]
K586		6,0000	1,9465	4,0000	1,2976	[2490]	[2505]	
K587		4,0000	1,2971	5,0000	1,6214	[4913]	[4915]	
K588		12,0000	6,2228	25,0000	12,9642	[5465]	[1019]	[4120] [5464] [1018] [4119]
K589		8,0000	2,5778	5,0000	1,6111	[7126]	[7128]	
K590		12,0000	5,8110	12,0000	5,8110	[9783]	[9791]	[9785] [9794] [9778] [9780]
K591		10,0000	3,2167	7,0000	2,2517	[9149]	[9166]	
K592		18,0000	7,2904	23,0000	9,3155	[6457]	[6251]	[6360]
K593		6,0000	2,5703	8,0000	3,4271	[702]	[703]	[726] [727]
K594		7,0000	2,2436	4,0000	1,2821	[433]	[434]	
K595		10,0000	3,2051	6,0000	1,9231	[3645]	[3646]	

Continua na próxima página ...

... continuação da página anterior									
Grupo	IT?	GSRank	GSIRank	GFRank	GFIrank	Métodos			
K596		5,0000	1,6015	4,0000	1,2812	[1452]	[1453]		
K597		5,0000	1,6015	4,0000	1,2812	[4248]	[4249]		
K598		5,0000	1,7840	7,0000	2,4976	[9746]	[9696]	[9699]	
K599		7,0000	2,2273	4,0000	1,2727	[8667]	[8670]		
K600		5,0000	1,5908	5,0000	1,5908	[3082]	[3084]		
K601		4,0000	1,2724	4,0000	1,2724	[5914]	[5924]		
K602		5,0000	1,5856	4,0000	1,2684	[5377]	[5379]		
K603		11,0000	4,6403	7,0000	2,9529	[6043]	[6044]	[6045]	
K604		8,0000	2,5289	6,0000	1,8967	[4407]	[4426]		
K605		7,0000	2,1990	4,0000	1,2566	[2476]	[2479]		
K606		8,0000	3,1452	10,0000	3,9315	[6257]	[6377]	[6256]	[6375]
K607		31,0000	14,0955	96,0000	43,6507	[8900]	[8924]	[8908]	[8931]
K608		6,0000	3,0523	15,0000	7,6307	[8058]	[8055]	[8048]	[8051]
K609		12,0000	5,3421	14,0000	6,2325	[6108]	[6104]	[6102]	[6103]
K610		7,0000	2,7486	8,0000	3,1413	[1256]	[1254]	[1326]	
K611		5,0000	1,9594	7,0000	2,7432	[2888]	[3072]	[3073]	
K612		27,0000	11,5817	35,0000	15,0133	[8161]	[8162]	[8165]	
K613		5,0000	1,5429	4,0000	1,2343	[1900]	[1906]		
K614		9,0000	2,7701	5,0000	1,5390	[3639]	[3640]		
K615		8,0000	3,1134	10,0000	3,8918	[6255]	[6376]	[6254]	[6374]
K616		8,0000	2,4495	5,0000	1,5310	[8222]	[8249]		
K617		7,0000	2,1389	4,0000	1,2222	[5407]	[5409]		
K618		6,0000	2,6042	14,0000	6,0765	[4701]	[4698]	[4696]	[4697]
K619		18,0000	5,4900	14,0000	4,2700	[3548]	[3549]		
K620		8,0000	2,6622	8,0000	2,6622	[9767]	[9764]	[9765]	
K621		6,0000	1,8283	4,0000	1,2189	[5548]	[5598]		
K622		9,0000	4,0915	10,0000	4,5461	[6109]	[6054]	[6041]	[6110]
K623		14,0000	5,6102	41,0000	16,4298	[6112]	[6145]	[6046]	[6113]
K624		10,0000	3,8776	12,0000	4,6531	[2373]	[3030]	[2374]	[3031]
K625		12,0000	3,5974	11,0000	3,2976	[9365]	[9368]		
K626		8,0000	2,3937	5,0000	1,4960	[8748]	[8750]		
K627		13,0000	3,8800	13,0000	3,8800	[9657]	[9660]		
K628		8,0000	2,3855	5,0000	1,4909	[2143]	[2151]		
K629	Sim	97,0000	30,5646	72,0000	22,6871	[7407]	[7406]	[7409]	
K630		7,0000	2,7259	9,0000	3,5048	[10199]	[9919]	[10014]	
K631		6,0000	2,2299	7,0000	2,6016	[2992]	[3069]	[3086]	
K632		7,0000	2,0533	4,0000	1,1733	[7308]	[7309]		
K633		7,0000	2,0474	4,0000	1,1699	[9290]	[9291]		
K634		4,0000	1,1690	4,0000	1,1690	[5794]	[6006]		
K635		12,0000	4,0291	14,0000	4,7006	[8517]	[8514]	[8515]	
K636		11,0000	4,6641	22,0000	9,3282	[8026]	[8027]	[8020]	[8059]
K637		6,0000	1,7383	5,0000	1,4486	[9168]	[9170]		
K638		5,0000	1,4484	4,0000	1,1587	[5092]	[5093]		
K639		5,0000	1,7455	6,0000	2,0945	[512]	[511]	[514]	
K640		8,0000	2,3382	10,0000	2,9227	[2645]	[2659]	[2660]	
K641		5,0000	1,7455	6,0000	2,0945	[3731]	[3730]	[3733]	
K642		6,0000	1,7321	5,0000	1,4435	[5380]	[5381]		
K643		4,0000	1,5808	6,0000	2,3711	[5200]	[7566]	[7571]	
K644		9,0000	3,6560	7,0000	2,8436	[230]	[231]	[232]	
K645		7,0000	2,3217	8,0000	2,6533	[5125]	[5132]	[5133]	
K646		6,0000	1,7225	4,0000	1,1484	[7991]	[8806]		
K647		11,0000	3,1549	6,0000	1,7209	[427]	[428]		
K648		4,0000	1,1467	4,0000	1,1467	[7598]	[7601]		
K649		4,0000	1,1467	4,0000	1,1467	[7676]	[7679]		
K650		4,0000	1,1467	4,0000	1,1467	[7712]	[7713]		
K651		13,0000	5,1437	13,0000	5,1437	[8178]	[8171]	[8170]	[8172]
K652		7,0000	2,4602	8,0000	2,8116	[6501]	[6502]	[6508]	[7070]
K653		21,0000	7,8889	16,0000	6,0106	[5442]	[5475]	[5424]	[5443]
K654		7,0000	1,9989	4,0000	1,1422	[6542]	[6545]		
K655		7,0000	1,9950	8,0000	2,2800	[10035]	[10036]		
K656		5,0000	1,4214	4,0000	1,1371	[5205]	[5207]		
K657		13,0000	4,7335	15,0000	5,4618	[6894]	[7003]	[6834]	[6962]
K658		7,0000	2,4702	6,0000	2,1173	[8121]	[8122]	[8123]	
K659		46,0000	17,4272	65,0000	24,6254	[8173]	[8169]	[8174]	
K660		18,0000	7,4282	12,0000	4,9521	[4307]	[4308]	[1531]	[1532]
K661		10,0000	2,8278	8,0000	2,2622	[857]	[860]		
K662		13,0000	5,1868	11,0000	4,3889	[6538]	[6539]	[6540]	
K663		5,0000	1,6602	8,0000	2,6564	[5189]	[7564]	[7572]	
K664		7,0000	2,5075	7,0000	2,5075	[1270]	[1267]	[1269]	

Na Tabela C.5 são apresentados os métodos do sistema HSQLDB com o respectivo *fan-in*. Devido a limitação de espaço, foram listados somente os métodos com *fan-in* a partir de 5.

Tabela C.5: Lista de métodos do HSQLDB

[Id] Método(Parâmetros)	<i>fan-in</i>
[13]boolean BaseMemoryNode.equals(Node)	6
[17]index.RowIterator BaseTable.rowIterator(Session)	8
[32]void CachedRow.delete()	5
[62]boolean Column.isIdentity()	8
[63]boolean Column.isNullable()	7
[65]boolean Column.isPrimaryKey()	6
[82]int Column.compare(Collation.Object,Object,int)	9
[83]Object Column.convertObject(Object,int)	21
[186]boolean Database.isFilesReadOnly()	9
[187]boolean Database.isFilesInJar()	7

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros) <span style="float: right;">fan-in</span>	
191	boolean Database.isReferentialIntegrity()	5
212	boolean Database.isStoredFileAccess()	6
218	int[] DatabaseCommandInterpreter.processColumnList(Table)	5
325	void DatabaseInformationMain.addColumn(Table,String,int,int,boolean)	20
326	void DatabaseInformationMain.addColumn(Table,String,int,boolean)	44
327	void DatabaseInformationMain.addColumn(Table,String,int)	38
328	lib.Iterator DatabaseInformationMain.allTables()	6
332	boolean DatabaseInformationMain.isAccessibleTable(Table)	17
333	Table DatabaseInformationMain.createBlankTable(HsqlNameManager\$HsqlName)	45
373	boolean DatabaseObjectNames.containsName(String)	5
413	int DIPcedureInfo.colOffset()	6
458	boolean DiskNode.equals(Node)	6
560	String Expression.describe(Session)	6
578	boolean Expression.isAggregate()	9
601	void Expression.resolveTables(TableFilter)	7
602	void Expression.resolveTypes(Session)	10
623	boolean Expression.testCondition(Session)	7
624	Boolean Expression.test(Session)	6
635	boolean Expression.isParam()	8
636	boolean Expression.isFixedConstant()	5
642	void Expression\$Collector.addAll(Expression,int)	9
697	void Grantee.updateAllRights()	6
705	void GranteeManager.updateAllRights(Grantee)	5
740	Result HSQLClientConnection.execute(Result)	18
759	java.sql.Timestamp HsqlDateTime.timestampValue(String)	5
762	java.sql.Date HsqlDateTime.dateValue(String)	5
763	java.sql.Time HsqlDateTime.timeValue(String)	5
776	void HsqlDateTime.resetToDate(Calendar)	7
809	HsqlNameManager.HsqlName HsqlNameManager.newHsqlName(String,boolean)	14
812	HsqlNameManager.HsqlName HsqlNameManager.newAutoName(String)	8
819	void HsqlNameManager\$HsqlName.rename(String,boolean)	7
822	int HsqlNameManager\$HsqlName.hashCode()	8
823	boolean HsqlNameManager\$HsqlName.isReservedIndexName(String)	5
854	Result HTTPClientConnection.execute(Result)	18
857	Row index.RowIterator.next()	16
858	boolean index.RowIterator.hasNext()	15
864	boolean Index.isUnique()	5
889	index.RowIterator Index.firstRow(Session)	6
901	boolean Index\$IndexRowIterator.hasNext()	15
902	Row Index\$IndexRowIterator.next()	16
919	int jdbc.jdbcCallableStatement.findParameterIndex(String)	51
922	void jdbc.jdbcCallableStatement.registerOutParameter(int,int)	5
1112	void jdbc.jdbcConnection.checkClosed()	34
1114	void jdbc.jdbcConnection.clearWarningsNoCheck()	7
1115	int jdbc.jdbcConnection.xlateRSType(int)	6
1116	int jdbc.jdbcConnection.xlateRSConcurrency(int)	6
1297	String jdbc.jdbcDatabaseMetaData.and(String,String,Object)	17
1298	java.sql.ResultSet jdbc.jdbcDatabaseMetaData.execute(String)	24
1299	java.sql.ResultSet jdbc.jdbcDatabaseMetaData.executeSelect(String,String)	15
1300	StringBuffer jdbc.jdbcDatabaseMetaData.toQueryPrefix(String)	17
1301	boolean jdbc.jdbcDatabaseMetaData.wantsIsNull(String)	10
1302	String jdbc.jdbcDatabaseMetaData.translateSchema(String)	17
1330	void jdbc.jdbcParameterMetaData.checkRange(int)	8
1385	void jdbc.jdbcPreparedStatement.checkSetParameterIndex(int)	10
1458	int jdbc.jdbcResultSet.findColumn(String)	46
1557	void jdbc.jdbcResultSet.checkClosed()	20
1640	void jdbc.jdbcResultSetMetaData.checkColumn(int)	20
1686	void jdbc.jdbcStatement.checkClosed()	36
1695	void jdbc.Util.throwError(Result)	9
1696	java.sql.SQLException jdbc.Util.sqlException(HsqlException)	30
1697	java.sql.SQLException jdbc.Util.sqlException(int)	31
1698	java.sql.SQLException jdbc.Util.sqlException(int,String)	33
1743	Object lib.ArrayUtil.resizeArray(Object,int)	7
1744	Object lib.ArrayUtil.toAdjustedArray(Object,Object,int,int)	5
1756	boolean lib.BaseList.add(Object)	32
1757	int lib.BaseList.size()	46
1766	boolean lib.BaseList\$BaseListIterator.hasNext()	88
1767	Object lib.BaseList\$BaseListIterator.next()	87
1777	int lib.Collection.size()	56
1779	boolean lib.Collection.contains(Object)	24
1780	lib.Iterator lib.Collection.iterator()	60
1781	boolean lib.Collection.add(Object)	55
1782	boolean lib.Collection.remove(Object)	8
1784	void lib.Collection.clear()	8
1791	int lib.DoubleIntIndex.size()	6
1814	void lib.DoubleIntIndex.moveRows(int,int,int)	5
1820	InputStream lib.FileAccess.openInputStreamElement(String)	6
1821	OutputStream lib.FileAccess.openOutputStreamElement(String)	6
1822	boolean lib.FileAccess.isStreamElement(String)	12
1824	void lib.FileAccess.removeElement(String)	11
1829	boolean lib.FileUtil.isStreamElement(String)	12
1830	InputStream lib.FileUtil.openInputStreamElement(String)	6
1832	void lib.FileUtil.removeElement(String)	11
1834	OutputStream lib.FileUtil.openOutputStreamElement(String)	6
1846	String lib.FileUtil.canonicalOrAbsolutePath(String)	5
1855	Object lib.HashMap.put(Object,Object)	21
1856	Object lib.HashMap.remove(Object)	15
1857	boolean lib.HashMap.containsKey(Object)	12
1860	lib.Set lib.HashMap.keySet()	14
1861	lib.Collection lib.HashMap.values()	13
1863	lib.Iterator lib.HashMap\$KeySet.iterator()	38

Continua na próxima página ...

... continuação da página anterior		
[Id] Método(Parâmetros)	fan-in	
1873	lib.Iterator lib.HashMap\$Values.iterator()	19
1885	Object lib.HashMappedList.remove(Object)	15
1886	Object lib.HashMappedList.remove(int)	6
1887	boolean lib.HashMappedList.add(Object, Object)	13
1888	Object lib.HashMappedList.put(Object, Object)	21
1895	void lib.HashMappedList.checkRange(int)	6
1899	boolean lib.HashSet.contains(Object)	24
1901	boolean lib.HashSet.add(Object)	25
1904	boolean lib.HashSet.remove(Object)	8
1906	lib.Iterator lib.HashSet.iterator()	60
1922	boolean lib.HsqlArrayList.add(Object)	29
1927	int lib.HsqlArrayList.size()	42
1930	void lib.HsqlArrayList.clear()	5
1933	Object lib.HsqlArrayList.toArray(Object)	11
1946	int lib.HsqlByteArrayInputStream.readInt()	10
1947	long lib.HsqlByteArrayInputStream.readLong()	7
1953	int lib.HsqlByteArrayInputStream.read()	7
1955	long lib.HsqlByteArrayInputStream.skip(long)	7
1959	void lib.HsqlByteArrayInputStream.reset()	8
1965	void lib.HsqlByteArrayOutputStream.writeInt(int)	10
1966	void lib.HsqlByteArrayOutputStream.writeLong(long)	7
1967	void lib.HsqlByteArrayOutputStream.writeBytes(String)	13
1976	void lib.HsqlByteArrayOutputStream.write(int)	5
1977	void lib.HsqlByteArrayOutputStream.write(byte)	12
1978	void lib.HsqlByteArrayOutputStream.write(byte, int, int)	7
1980	void lib.HsqlByteArrayOutputStream.reset()	20
1982	int lib.HsqlByteArrayOutputStream.size()	23
1988	void lib.HsqlByteArrayOutputStream.ensureRoom(int)	15
2026	boolean lib.HsqlList.add(Object)	31
2031	int lib.HsqlList.size()	45
2091	Object lib.IntKeyHashMap.put(int, Object)	8
2093	Object lib.IntKeyHashMap.remove(int)	6
2095	lib.Collection lib.IntKeyHashMap.values()	5
2097	lib.Iterator lib.IntKeyHashMap\$KeySet.iterator()	38
2107	lib.Iterator lib.IntKeyHashMap\$Values.iterator()	19
2126	lib.Iterator lib.IntKeyInt ValueHashMap\$KeySet.iterator()	38
2136	lib.Iterator lib.IntKeyInt ValueHashMap\$Values.iterator()	19
2161	boolean lib.IntValueHashMap.put(Object, int)	6
2162	boolean lib.IntValueHashMap.remove(Object)	5
2163	boolean lib.IntValueHashMap.containsKey(Object)	5
2165	lib.Set lib.IntValueHashMap.keySet()	6
2167	lib.Iterator lib.IntValueHashMap\$KeySet.iterator()	38
2177	boolean lib.Iterator.hasNext()	93
2178	Object lib.Iterator.next()	88
2181	void lib.Iterator.remove()	5
2205	lib.Iterator lib.LongKeyInt ValueHashMap\$KeySet.iterator()	38
2215	lib.Iterator lib.LongKeyInt ValueHashMap\$Values.iterator()	19
2228	int lib.Set.size()	7
2230	boolean lib.Set.contains(Object)	24
2231	lib.Iterator lib.Set.iterator()	60
2232	boolean lib.Set.add(Object)	25
2234	boolean lib.Set.remove(Object)	8
2244	void lib.SimpleLog.logContext(Throwable)	15
2259	String lib.StopWatch.elapsedTimeToMessage(String)	5
2263	void lib.Storage.seek(long)	14
2268	int lib.Storage.readInt()	6
2272	void lib.Storage.close()	7
2278	String lib.StringConverter.byteToHex(byte)	6
2290	String lib.StringConverter.toQuotedString(String, char, boolean)	8
2303	boolean lib.StringUtil.isEmpty(String)	5
2313	boolean lib WrapperIterator.hasNext()	88
2314	Object lib WrapperIterator.next()	87
2430	boolean Node.equals(Node)	7
2439	long NumberSequence.peek()	6
2442	void NumberSequence.reset(long)	5
2452	SubQuery Parser.parseSubquery(int, HsqlNameManager\$HsqlName, boolean, int)	5
2454	Select Parser.parseSelect(int, boolean, boolean, boolean, boolean)	6
2465	Expression Parser.parseExpression()	9
2467	Expression Parser.readOr()	15
2474	void Parser.readThis(int)	17
2478	Expression Parser.readTerm()	6
2493	void Parser.read()	27
2496	void Parser.clearParameters()	5
2505	int Parser.parseOpenBrackets(Tokenizer)	11
2543	void persist.DataFileCache.open(boolean)	6
2544	void persist.DataFileCache.close(boolean)	10
2559	void persist.DataFileCache.saveRow(persist.CachedObject)	5
2612	boolean persist.HsqlProperties.isPropertyTrue(String)	16
2652	void persist.LockFile.trace(Object)	17
2660	void persist.Log.checkpoint(boolean)	8
2695	void persist.Logger.writeToLog(Session, String)	10
2696	void persist.Logger.writeInsertStatement(Session, Table, Object)	5
2720	void persist.NIOScaledRAFile.enlargeBuffer(long, int)	5
2721	void persist.NIOScaledRAFile.seek(long)	14
2725	int persist.NIOScaledRAFile.readInt()	5
2730	void persist.NIOScaledRAFile.close()	7
2745	void persist.ScaledRAFile.seek(long)	14
2750	int persist.ScaledRAFile.readInt()	5
2755	void persist.ScaledRAFile.close()	7
2760	void persist.ScaledRAFileInJar.seek(long)	10
2782	void persist.TextCache.open(boolean)	6

Continua na próxima página ...



... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
2784	void persist.TextCache.close(boolean)	10
2790	void persist.TextCache.saveRow(persist.CachedObject)	5
2847	void Result.add(Object)	14
2850	void Result.removeDuplicates(Session,int)	6
2857	void Result.write(rowio.RowOutputBinary)	5
2861	Result Result.read(rowio.RowInputBinary,DataInput)	7
2881	lib.Iterator Result.iterator()	7
2883	boolean Result\$ResultIterator.hasNext()	89
2884	Object Result\$ResultIterator.next()	87
2907	void Row.delete()	5
2929	int rowio.RowInputBase.readIntData()	6
2932	String rowio.RowInputBase.readString()	19
2957	int rowio.RowInputBinary.readIntData()	6
2959	String rowio.RowInputBinary.readString()	5
2978	String rowio.RowInputInterface.readString()	19
2980	int rowio.RowInputInterface.readIntData()	7
2988	String rowio.RowInputText.readString()	13
3010	void rowio.RowInputText.reset()	8
3016	String rowio.RowInputTextLog.readField()	7
3017	String rowio.RowInputTextLog.readNumberField()	8
3043	void rowio.RowOutputBase.writeSize(int)	10
3046	void rowio.RowOutputBase.writeIntData(int)	10
3047	void rowio.RowOutputBase.writeIntData(int,int)	5
3048	void rowio.RowOutputBase.writeString(String)	21
3065	void rowio.RowOutputBase.writeData(Object,Table)	6
3072	void rowio.RowOutputBinary.writeIntData(int)	9
3073	void rowio.RowOutputBinary.writeIntData(int,int)	5
3076	void rowio.RowOutputBinary.writeSize(int)	9
3078	void rowio.RowOutputBinary.writeString(String)	6
3097	void rowio.RowOutputBinary.ensureRoom(int)	15
3098	void rowio.RowOutputBinary.reset()	16
3102	void rowio.RowOutputInterface.writeSize(int)	10
3104	void rowio.RowOutputInterface.writeString(String)	21
3106	void rowio.RowOutputInterface.writeIntData(int)	10
3107	void rowio.RowOutputInterface.writeIntData(int,int)	5
3110	void rowio.RowOutputInterface.writeData(Object,Table)	6
3114	void rowio.RowOutputInterface.reset()	19
3119	void rowio.RowOutputText.writeSize(int)	5
3121	void rowio.RowOutputText.writeString(String)	11
3163	void rowio.RowOutputTextLog.writeString(String)	6
3172	void rowio.RowOutputTextLog.reset()	11
3211	lib.Iterator SchemaManager.userSchemaNameIterator()	5
3221	lib.Iterator SchemaManager.allTablesIterator()	17
3230	Table SchemaManager.findUserTable(Session,String,String)	9
3245	void SchemaManager.registerConstraintName(String,HsqlNameManager\$HsqlName)	6
3263	void SchemaManager.recompileViews(Table)	6
3312	void scriptio.ScriptWriterBase.sync()	8
3314	long scriptio.ScriptWriterBase.size()	5
3325	void scriptio.ScriptWriterBase.addSessionId(Session)	5
3353	void scriptio.ScriptWriterText.addSessionId(Session)	5
3364	void scriptio.ScriptWriterZipped.sync()	8
3377	void Select.prepareResult(Session)	5
3408	void Server.checkRunning(boolean)	8
3431	boolean Server.isSilent()	5
3457	void Server.print(String)	10
3459	void Server.printStackTrace(Throwable)	6
3461	void Server.printWithThread(String)	30
3462	void Server.printError(String)	5
3499	void Session.close()	6
3512	void Session.checkAdmin()	14
3515	void Session.checkDDLWrite()	11
3519	void Session.commit()	22
3530	boolean Session.isAutoCommit()	8
3540	Result Session.execute(Result)	13
3542	Result Session.sqlExecuteDirectNoPreChecks(String)	12
3571	Result SessionInterface.execute(Result)	21
3572	void SessionInterface.close()	6
3576	boolean SessionInterface.isAutoCommit()	8
3580	void SessionInterface.commit()	22
3608	Object store.BaseHashMap.addOrRemove(long,long,Object,Object,boolean)	15
3611	boolean store.BaseHashMap.reset()	8
3620	void store.BaseHashMap.clear()	28
3623	void store.BaseHashMap.resetAccessCount()	8
3624	int store.BaseHashMap.size()	53
3625	boolean store.BaseHashMap.isEmpty()	5
3626	boolean store.BaseHashMap.containsKey(Object)	14
3632	boolean store.BaseHashMap\$BaseHashIterator.hasNext()	92
3633	Object store.BaseHashMap\$BaseHashIterator.next()	88
3636	void store.BaseHashMap\$BaseHashIterator.remove()	5
3654	int store.HashIndex.linkNode(int,int)	8
3692	boolean Table.isText()	7
3695	boolean Table.isView()	17
3714	void Table.addColumn(Column)	6
3723	Table Table.moveDefinition(int,Column,int,int)	5
3729	void Table.updateConstraintsTables(Session,Table,int,int)	6
3740	int Table.findColumn(String)	6
3744	boolean Table.hasPrimaryKey()	11
3758	void Table.createPrimaryKey(int)	7
3771	boolean Table.isEmpty(Session)	10
3779	void Table.moveData(Session,Table,int,int)	6
3787	int Table.insertSys(Result)	6

Continua na próxima página ...

... continuação da página anterior		
[Id]	Método(Parâmetros)	fan-in
3791	void Table.insertSys(Object)	34
3793	void Table.enforceNullConstraints(Object)	5
3833	void Table.indexRow(Session,Row)	6
3834	void Table.clearAllRows(Session)	6
3841	Row Table.newRow(Object)	5
3904	void Tokenizer.back()	22
3907	void Tokenizer.throwUnexpected()	5
3908	boolean Tokenizer.isGetThis(String)	27
3910	boolean Tokenizer.wasQuotedIdentifier()	18
3913	boolean Tokenizer.wasSimpleName()	6
3938	HsqlException Trace.error(int,int,Object)	33
3939	HsqlException Trace.error(int,Object)	137
3943	HsqlException Trace.error(Result)	6
3945	RuntimeException Trace.runtimeError(int,Object)	5
3948	HsqlException Trace.error(int)	113
3949	void Trace.check(boolean,int)	16
3950	void Trace.check(boolean,int,Object)	15
3952	void Trace.printSystemOut(String)	20
3958	void Trace.doAssert(boolean)	22
3959	void Trace.doAssert(boolean,String)	28
4013	boolean Types.acceptsPrecisionCreateParam(int)	5
4015	boolean Types.acceptsScaleCreateParam(int)	5
4069	void util.CommonSwing.errorMessage(Exception)	18
4162	void util.DatabaseManager.refreshTree()	5
4167	String util.DatabaseManagerCommon.readFile(String)	6
4238	void util.Grid.addRow(String)	5
4371	void util.SqlFile.stdprintln(String)	9
4616	void util.ZaurusDatabaseManager.refreshTree()	5

Na Tabela C.6 são apresentados os grupos obtidos pela instância 12 da fase de agrupamento no sistema HSQLDB com o respectivo resultado do cálculo de pontuação de cada uma das medidas de ordenação. A Coluna IT? indica se o grupo contém métodos referentes a instâncias de interesses transversais validadas, conforme Tabela A.4. A Coluna Métodos apresenta os identificadores dos métodos que compõem cada grupo, sendo que os métodos podem ser consultados na Tabela C.5.

Tabela C.6: Lista de grupos da instância 12 com pontuação: HSQLDB

Grupo	IT?	GSRank	GSIRank	GFIRank	GFIRank	Métodos
K1		4,0000	4,0000	2,0000	2,0000	[9]
K2		13,0000	13,0000	8,0000	8,0000	[17]
K3		5,0000	5,0000	3,0000	3,0000	[56]
K4		4,0000	4,0000	2,0000	2,0000	[60]
K5		13,0000	13,0000	9,0000	9,0000	[82]
K6		30,0000	30,0000	21,0000	21,0000	[83]
K7		4,0000	4,0000	2,0000	2,0000	[86]
K8		6,0000	6,0000	3,0000	3,0000	[89]
K9		3,0000	3,0000	2,0000	2,0000	[93]
K10		7,0000	7,0000	4,0000	4,0000	[97]
K11		7,0000	7,0000	4,0000	4,0000	[98]
K12		5,0000	5,0000	3,0000	3,0000	[122]
K13		5,0000	5,0000	3,0000	3,0000	[140]
K14		5,0000	5,0000	2,0000	2,0000	[164]
K15		5,0000	5,0000	3,0000	3,0000	[168]
K16		4,0000	4,0000	2,0000	2,0000	[169]
K17		4,0000	4,0000	2,0000	2,0000	[183]
K18	Sim	17,0000	17,0000	9,0000	9,0000	[186]
K19		14,0000	14,0000	7,0000	7,0000	[187]
K20		8,0000	8,0000	5,0000	5,0000	[191]
K21		5,0000	5,0000	2,0000	2,0000	[195]
K22		7,0000	7,0000	4,0000	4,0000	[199]
K23		10,0000	10,0000	6,0000	6,0000	[212]
K24		4,0000	4,0000	2,0000	2,0000	[214]
K25		3,0000	3,0000	2,0000	2,0000	[223]
K26		3,0000	3,0000	2,0000	2,0000	[226]
K27		3,0000	3,0000	2,0000	2,0000	[258]
K28		7,0000	7,0000	6,0000	6,0000	[328]
K29		3,0000	3,0000	2,0000	2,0000	[362]
K30		5,0000	5,0000	4,0000	4,0000	[371]
K31		5,0000	5,0000	3,0000	3,0000	[377]
K32		5,0000	5,0000	3,0000	3,0000	[378]
K33		4,0000	4,0000	3,0000	3,0000	[390]
K34		7,0000	7,0000	3,0000	3,0000	[393]
K35		7,0000	7,0000	4,0000	4,0000	[396]
K36		3,0000	3,0000	2,0000	2,0000	[411]
K37		4,0000	4,0000	3,0000	3,0000	[447]
K38		4,0000	4,0000	2,0000	2,0000	[540]
K39		5,0000	5,0000	4,0000	4,0000	[559]
K40		3,0000	3,0000	2,0000	2,0000	[563]
K41		4,0000	4,0000	2,0000	2,0000	[567]
K42		12,0000	12,0000	9,0000	9,0000	[578]
K43		4,0000	4,0000	2,0000	2,0000	[584]
K44		6,0000	6,0000	3,0000	3,0000	[596]
K45		8,0000	8,0000	6,0000	6,0000	[624]

Continua na próxima página ...

... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K46		4,0000	4,0000	2,0000	2,0000	[634]
K47		12,0000	12,0000	8,0000	8,0000	[635]
K48		13,0000	13,0000	9,0000	9,0000	[642]
K49		5,0000	5,0000	2,0000	2,0000	[679]
K50	Sim	6,0000	6,0000	4,0000	4,0000	[686]
K51	Sim	5,0000	5,0000	3,0000	3,0000	[690]
K52		4,0000	4,0000	2,0000	2,0000	[717]
K53		4,0000	4,0000	2,0000	2,0000	[746]
K54		9,0000	9,0000	7,0000	7,0000	[776]
K55		4,0000	4,0000	2,0000	2,0000	[777]
K56		4,0000	4,0000	2,0000	2,0000	[793]
K57		20,0000	20,0000	14,0000	14,0000	[809]
K58		9,0000	9,0000	4,0000	4,0000	[821]
K59		12,0000	12,0000	8,0000	8,0000	[822]
K60		9,0000	9,0000	5,0000	5,0000	[823]
K61		9,0000	9,0000	5,0000	5,0000	[864]
K62		5,0000	5,0000	3,0000	3,0000	[875]
K63		6,0000	6,0000	3,0000	3,0000	[882]
K64		6,0000	6,0000	5,0000	5,0000	[922]
K65		6,0000	6,0000	3,0000	3,0000	[1075]
K66		4,0000	4,0000	3,0000	3,0000	[1113]
K67		10,0000	10,0000	7,0000	7,0000	[1114]
K68		3,0000	3,0000	2,0000	2,0000	[1559]
K69		4,0000	4,0000	3,0000	3,0000	[1616]
K70		4,0000	4,0000	3,0000	3,0000	[1687]
K71		14,0000	14,0000	9,0000	9,0000	[1695]
K72		3,0000	3,0000	2,0000	2,0000	[1711]
K73		6,0000	6,0000	3,0000	3,0000	[1714]
K74		6,0000	6,0000	3,0000	3,0000	[1718]
K75		4,0000	4,0000	2,0000	2,0000	[1723]
K76		4,0000	4,0000	2,0000	2,0000	[1727]
K77		5,0000	5,0000	2,0000	2,0000	[1734]
K78		7,0000	7,0000	4,0000	4,0000	[1735]
K79		8,0000	8,0000	4,0000	4,0000	[1739]
K80		14,0000	14,0000	7,0000	7,0000	[1743]
K81		4,0000	4,0000	2,0000	2,0000	[1745]
K82		5,0000	5,0000	2,0000	2,0000	[1749]
K83		3,0000	3,0000	2,0000	2,0000	[1760]
K84		5,0000	5,0000	2,0000	2,0000	[1761]
K85		5,0000	5,0000	2,0000	2,0000	[1762]
K86		4,0000	4,0000	2,0000	2,0000	[1764]
K87		9,0000	9,0000	6,0000	6,0000	[1791]
K88		5,0000	5,0000	3,0000	3,0000	[1802]
K89		6,0000	6,0000	5,0000	5,0000	[1814]
K90		5,0000	5,0000	4,0000	4,0000	[1817]
K91		3,0000	3,0000	2,0000	2,0000	[1819]
K92		6,0000	6,0000	4,0000	4,0000	[1835]
K93		5,0000	5,0000	2,0000	2,0000	[1836]
K94		7,0000	7,0000	4,0000	4,0000	[1837]
K95		5,0000	5,0000	3,0000	3,0000	[1840]
K96		9,0000	9,0000	5,0000	5,0000	[1846]
K97		8,0000	8,0000	4,0000	4,0000	[1847]
K98		23,0000	23,0000	14,0000	14,0000	[1860]
K99		22,0000	22,0000	13,0000	13,0000	[1861]
K100		11,0000	11,0000	6,0000	6,0000	[1886]
K101		20,0000	20,0000	13,0000	13,0000	[1887]
K102		8,0000	8,0000	4,0000	4,0000	[1905]
K103		4,0000	4,0000	3,0000	3,0000	[1928]
K104		18,0000	18,0000	11,0000	11,0000	[1933]
K105		5,0000	5,0000	3,0000	3,0000	[1943]
K106		14,0000	14,0000	10,0000	10,0000	[1946]
K107		10,0000	10,0000	7,0000	7,0000	[1955]
K108		17,0000	17,0000	12,0000	12,0000	[1977]
K109		5,0000	5,0000	2,0000	2,0000	[1991]
K110		6,0000	6,0000	3,0000	3,0000	[2003]
K111		3,0000	3,0000	2,0000	2,0000	[2005]
K112		4,0000	4,0000	2,0000	2,0000	[2015]
K113		6,0000	6,0000	4,0000	4,0000	[2020]
K114		5,0000	5,0000	4,0000	4,0000	[2067]
K115		6,0000	6,0000	4,0000	4,0000	[2070]
K116		5,0000	5,0000	2,0000	2,0000	[2074]
K117		4,0000	4,0000	2,0000	2,0000	[2075]
K118		6,0000	6,0000	3,0000	3,0000	[2084]
K119		9,0000	9,0000	5,0000	5,0000	[2095]
K120		4,0000	4,0000	2,0000	2,0000	[2152]
K121		11,0000	11,0000	6,0000	6,0000	[2165]
K122		7,0000	7,0000	3,0000	3,0000	[2183]
K123		8,0000	8,0000	4,0000	4,0000	[2185]
K124		4,0000	4,0000	2,0000	2,0000	[2189]
K125		13,0000	13,0000	7,0000	7,0000	[2228]
K126		4,0000	4,0000	2,0000	2,0000	[2242]
K127		5,0000	5,0000	2,0000	2,0000	[2243]
K128		20,0000	20,0000	15,0000	15,0000	[2244]
K129		6,0000	6,0000	3,0000	3,0000	[2253]
K130		8,0000	8,0000	5,0000	5,0000	[2259]
K131		4,0000	4,0000	2,0000	2,0000	[2277]
K132		11,0000	11,0000	6,0000	6,0000	[2278]
K133		6,0000	6,0000	3,0000	3,0000	[2279]
K134		4,0000	4,0000	2,0000	2,0000	[2282]
K135		6,0000	6,0000	3,0000	3,0000	[2287]
K136		5,0000	5,0000	2,0000	2,0000	[2297]

Continua na próxima página ...

... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K137		8,0000	8,0000	5,0000	5,0000	[2303]
K138		3,0000	3,0000	2,0000	2,0000	[2327]
K139		3,0000	3,0000	2,0000	2,0000	[2398]
K140		9,0000	9,0000	4,0000	4,0000	[2414]
K141		12,0000	12,0000	6,0000	6,0000	[2439]
K142		9,0000	9,0000	5,0000	5,0000	[2442]
K143		3,0000	3,0000	2,0000	2,0000	[2450]
K144		8,0000	8,0000	6,0000	6,0000	[2454]
K145		3,0000	3,0000	2,0000	2,0000	[2463]
K146		12,0000	12,0000	9,0000	9,0000	[2465]
K147		7,0000	7,0000	6,0000	6,0000	[2478]
K148		5,0000	5,0000	3,0000	3,0000	[2512]
K149		6,0000	6,0000	3,0000	3,0000	[2557]
K150		4,0000	4,0000	2,0000	2,0000	[2572]
K151		27,0000	27,0000	16,0000	16,0000	[2612]
K152		6,0000	6,0000	3,0000	3,0000	[2613]
K153		4,0000	4,0000	2,0000	2,0000	[2617]
K154		6,0000	6,0000	3,0000	3,0000	[2623]
K155		4,0000	4,0000	3,0000	3,0000	[2643]
K156		10,0000	10,0000	8,0000	8,0000	[2660]
K157		6,0000	6,0000	4,0000	4,0000	[2675]
K158		4,0000	4,0000	2,0000	2,0000	[2684]
K159		5,0000	5,0000	2,0000	2,0000	[2694]
K160		6,0000	6,0000	5,0000	5,0000	[2720]
K161		8,0000	8,0000	4,0000	4,0000	[2742]
K162		4,0000	4,0000	3,0000	3,0000	[2747]
K163		3,0000	3,0000	2,0000	2,0000	[2773]
K164		9,0000	9,0000	4,0000	4,0000	[2831]
K165		4,0000	4,0000	2,0000	2,0000	[2844]
K166		24,0000	24,0000	14,0000	14,0000	[2847]
K167		4,0000	4,0000	2,0000	2,0000	[2849]
K168		5,0000	5,0000	2,0000	2,0000	[2856]
K169		14,0000	14,0000	7,0000	7,0000	[2861]
K170		14,0000	14,0000	7,0000	7,0000	[2881]
K171		5,0000	5,0000	3,0000	3,0000	[2816]
K172		6,0000	6,0000	4,0000	4,0000	[2818]
K173		6,0000	6,0000	4,0000	4,0000	[2909]
K174		7,0000	7,0000	3,0000	3,0000	[2946]
K175		5,0000	5,0000	2,0000	2,0000	[2982]
K176		5,0000	5,0000	2,0000	2,0000	[2983]
K177		4,0000	4,0000	2,0000	2,0000	[2992]
K178		4,0000	4,0000	2,0000	2,0000	[3009]
K179		3,0000	3,0000	2,0000	2,0000	[3126]
K180		9,0000	9,0000	6,0000	6,0000	[3163]
K181		4,0000	4,0000	2,0000	2,0000	[3208]
K182		6,0000	6,0000	3,0000	3,0000	[3220]
K183		13,0000	13,0000	9,0000	9,0000	[3230]
K184		3,0000	3,0000	2,0000	2,0000	[3295]
K185		4,0000	4,0000	2,0000	2,0000	[3307]
K186		7,0000	7,0000	5,0000	5,0000	[3314]
K187		3,0000	3,0000	2,0000	2,0000	[3354]
K188		9,0000	9,0000	5,0000	5,0000	[3377]
K189		9,0000	9,0000	8,0000	8,0000	[3408]
K190		7,0000	7,0000	5,0000	5,0000	[3431]
K191		12,0000	12,0000	10,0000	10,0000	[3457]
K192		20,0000	20,0000	12,0000	12,0000	[3542]
K193		5,0000	5,0000	4,0000	4,0000	[3569]
K194		5,0000	5,0000	3,0000	3,0000	[3612]
K195		5,0000	5,0000	3,0000	3,0000	[3617]
K196		48,0000	48,0000	28,0000	28,0000	[3620]
K197		84,0000	84,0000	53,0000	53,0000	[3624]
K198		10,0000	10,0000	5,0000	5,0000	[3625]
K199		3,0000	3,0000	2,0000	2,0000	[3646]
K200		5,0000	5,0000	3,0000	3,0000	[3655]
K201		4,0000	4,0000	2,0000	2,0000	[3663]
K202		6,0000	6,0000	3,0000	3,0000	[3686]
K203		6,0000	6,0000	3,0000	3,0000	[3689]
K204	Sim	24,0000	24,0000	17,0000	17,0000	[3695]
K205		3,0000	3,0000	2,0000	2,0000	[3701]
K206		3,0000	3,0000	2,0000	2,0000	[3731]
K207		9,0000	9,0000	6,0000	6,0000	[3740]
K208		17,0000	17,0000	11,0000	11,0000	[3744]
K209		5,0000	5,0000	3,0000	3,0000	[3749]
K210		10,0000	10,0000	7,0000	7,0000	[3758]
K211		5,0000	5,0000	2,0000	2,0000	[3759]
K212		7,0000	7,0000	4,0000	4,0000	[3762]
K213	Sim	16,0000	16,0000	10,0000	10,0000	[3771]
K214		4,0000	4,0000	3,0000	3,0000	[3795]
K215		5,0000	5,0000	2,0000	2,0000	[3801]
K216		11,0000	11,0000	6,0000	6,0000	[3834]
K217		5,0000	5,0000	2,0000	2,0000	[3836]
K218		4,0000	4,0000	2,0000	2,0000	[3874]
K219		5,0000	5,0000	4,0000	4,0000	[3885]
K220		7,0000	7,0000	3,0000	3,0000	[3903]
K221		32,0000	32,0000	27,0000	27,0000	[3908]
K222		10,0000	10,0000	6,0000	6,0000	[3943]
K223		9,0000	9,0000	5,0000	5,0000	[3945]
K224		24,0000	24,0000	16,0000	16,0000	[3949]
K225	Sim	31,0000	31,0000	20,0000	20,0000	[3952]
K226		5,0000	5,0000	3,0000	3,0000	[3967]
K227		4,0000	4,0000	2,0000	2,0000	[3986]

Continua na próxima página ...

... continuação da página anterior						
Grupo	IT?	GSIRank	GSIRank	GFRank	GFIRank	Métodos
K228		7,0000	7,0000	4,0000	4,0000	[4052]
K229		4,0000	4,0000	2,0000	2,0000	[4054]
K230		4,0000	4,0000	2,0000	2,0000	[4068]
K231	Sim	22,0000	22,0000	18,0000	18,0000	[4069]
K232		4,0000	4,0000	2,0000	2,0000	[4070]
K233		3,0000	3,0000	2,0000	2,0000	[4077]
K234		3,0000	3,0000	2,0000	2,0000	[4109]
K235		3,0000	3,0000	2,0000	2,0000	[4164]
K236		5,0000	5,0000	2,0000	2,0000	[4169]
K237		3,0000	3,0000	2,0000	2,0000	[4197]
K238	Sim	3,0000	3,0000	2,0000	2,0000	[4198]
K239		8,0000	8,0000	5,0000	5,0000	[4238]
K240		5,0000	5,0000	4,0000	4,0000	[4240]
K241		5,0000	5,0000	3,0000	3,0000	[4264]
K242		5,0000	5,0000	2,0000	2,0000	[4265]
K243		3,0000	3,0000	2,0000	2,0000	[4341]
K244		3,0000	3,0000	2,0000	2,0000	[4365]
K245		3,0000	3,0000	2,0000	2,0000	[4379]
K246		3,0000	3,0000	2,0000	2,0000	[4385]
K247		3,0000	3,0000	2,0000	2,0000	[4414]
K248		5,0000	5,0000	4,0000	4,0000	[4440]
K249		4,0000	4,0000	2,0000	2,0000	[4565]
K250		4,0000	4,0000	3,0000	3,0000	[4568]
K251		4,0000	4,0000	2,0000	2,0000	[4666]
K252		4,0000	4,0000	2,0000	2,0000	[4671]
K253		4,0000	4,0000	3,0000	3,0000	[4688]
K254		4,0000	2,9111	4,0000	2,9111	[3316] [3365]
K255		14,0000	10,2489	15,0000	10,9810	[759] [762] [763]
K256		26,0000	17,9785	30,0000	20,7444	[1856] [1885]
K257		4,0000	2,7127	4,0000	2,7127	[2645] [2716]
K258		7,0000	5,1333	15,0000	11,0000	[1117] [1115] [1116]
K259		14,0000	9,4176	20,0000	13,4537	[2544] [2784]
K260		8,0000	5,2789	10,0000	6,5986	[2559] [2790]
K261		37,0000	23,9332	42,0000	27,1674	[1855] [1888]
K262		8,0000	4,9867	9,0000	5,6100	[636] [637]
K263		42,0000	26,0482	50,0000	31,0097	[1901] [2232]
K264		9,0000	5,5596	12,0000	7,4129	[2543] [2782]
K265		19,0000	11,7306	30,0000	18,5220	[1988] [3097]
K266		34,0000	20,8100	27,0000	16,5256	[3098] [3172]
K267		28,0000	16,9400	44,0000	26,6200	[3519] [3580]
K268		6,0000	3,8125	15,0000	9,5312	[2502] [2504] [2503] [2499] [2500]
K269		4,0000	2,3380	4,0000	2,3380	[3525] [3526]
K270		7,0000	4,0839	6,0000	3,5005	[2582] [2618]
K271		19,0000	11,0309	24,0000	13,9337	[1822] [1829]
K272		40,0000	23,6259	72,0000	42,5267	[1779] [1899] [2230]
K273		6,0000	3,4264	5,0000	2,8554	[3313] [3315]
K274		20,0000	11,3818	14,0000	7,9673	[2091] [2093]
K275		7,0000	3,9526	8,0000	4,5173	[2584] [2619]
K276		12,0000	6,5743	12,0000	6,5743	[3499] [3572]
K277		17,0000	9,2945	22,0000	12,0282	[1824] [1832]
K278		7,0000	4,1122	9,0000	5,2871	[1649] [920] [1393]
K279		10,0000	5,4000	5,0000	2,7000	[2621] [2622]
K280		124,0000	75,5032	887,0000	540,0915	[1766] [2883] [2177] [3632] [2313] [2314] [1767] [3633] [2178] [2884]
K281		15,0000	8,0603	16,0000	8,5976	[3530] [3576]
K282		8,0000	4,2844	10,0000	5,3556	[32] [2907]
K283		4,0000	2,1077	4,0000	2,1077	[2542] [2778]
K284		11,0000	5,7470	10,0000	5,2245	[4013] [4015]
K285		14,0000	7,3033	16,0000	8,3467	[1959] [3010]
K286		6,0000	3,1157	4,0000	2,0771	[3516] [3517]
K287		13,0000	6,7427	12,0000	6,2240	[1820] [1830]
K288		13,0000	6,7427	12,0000	6,2240	[1821] [1834]
K289		17,0000	8,9814	24,0000	12,6796	[1782] [1904] [2234]
K290		13,0000	6,7072	7,0000	3,6116	[3211] [3218]
K291		5,0000	2,5725	4,0000	2,0580	[2046] [2307]
K292		5,0000	2,5703	6,0000	3,0844	[2546] [2779]
K293		4,0000	2,0533	4,0000	2,0533	[3323] [3356]
K294		5,0000	2,5623	4,0000	2,0499	[831] [840]
K295		5,0000	2,5607	4,0000	2,0486	[4165] [4166]
K296		4,0000	2,0349	4,0000	2,0349	[3050] [3131]
K297		5,0000	2,5319	4,0000	2,0255	[2563] [2564]
K298		6,0000	3,0380	6,0000	3,0380	[2685] [2686]
K299		8,0000	4,0400	8,0000	4,0400	[3520] [3581]
K300		8,0000	4,0343	6,0000	3,0257	[3432] [3433]
K301		5,0000	3,2831	6,0000	3,9398	[3629] [1858] [2092]
K302		20,0000	10,0278	25,0000	12,5347	[3512] [3515]
K303		6,0000	3,0060	6,0000	3,0060	[2556] [2787]
K304		6,0000	2,9973	6,0000	2,9973	[1989] [3099]
K305		16,0000	7,9324	15,0000	7,4367	[3016] [3017]
K306		7,0000	3,4644	11,0000	5,4440	[3833] [3841]
K307		6,0000	2,9550	6,0000	2,9550	[3277] [3302]
K308		10,0000	4,8833	9,0000	4,3950	[2696] [2697]
K309		9,0000	4,3470	9,0000	4,3470	[452] [458]
K310		6,0000	2,8617	6,0000	2,8617	[3835] [3895]
K311	Sim	38,0000	18,0817	34,0000	16,1783	[1694] [1696]
K312		7,0000	3,3160	5,0000	2,3686	[595] [3375]
K313		8,0000	4,0059	12,0000	6,0089	[3355] [3326] [3343]
K314	Sim	9,0000	4,2586	8,0000	3,7855	[3528] [3574]
K315		9,0000	4,7039	12,0000	6,2718	[2271] [2729] [2754]
K316		8,0000	3,7156	6,0000	2,7867	[3704] [3888]
K317		12,0000	5,5714	13,0000	6,0357	[13] [2430]
K318		5,0000	2,3071	5,0000	2,3071	[2762] [2774]

Continua na próxima página ...

... continuação da página anterior							
Grupo	IT?	GSIRank	GSIRank	GFRank	GFIRank	Métodos	
K319	Sim	6,0000	2,7646	9,0000	4,1468	[3793]	[3796]
K320		5,0000	2,3023	6,0000	2,7627	[3124]	[3174]
K321		8,0000	3,6800	5,0000	2,3000	[3659]	[3660]
K322		42,0000	21,1991	37,0000	18,6754	[1967]	[1964] [1976] [1965] [1978]
K323		7,0000	3,2044	4,0000	1,8311	[699]	[700]
K324		10,0000	4,5667	8,0000	3,6533	[2850]	[3387]
K325		12,0000	5,4686	7,0000	3,1900	[878]	[881]
K326		5,0000	2,2730	4,0000	1,8184	[3066]	[3111]
K327		28,0000	14,9764	62,0000	33,1621	[858]	[901] [857] [902]
K328		26,0000	14,6494	31,0000	17,4666	[2163]	[1857] [3626]
K329		8,0000	3,6160	8,0000	3,6160	[1825]	[1833]
K330		5,0000	2,2508	6,0000	2,7010	[3721]	[3894]
K331		5,0000	2,2508	4,0000	1,8006	[3817]	[3893]
K332		4,0000	1,7988	4,0000	1,7988	[2690]	[2712]
K333		8,0000	4,6993	12,0000	7,0489	[2416]	[3] [2412] [2802]
K334		7,0000	3,3478	6,0000	2,8696	[4084]	[4083] [4086]
K335		6,0000	2,6400	6,0000	2,6400	[4088]	[4091]
K336		12,0000	6,0632	10,0000	5,0526	[3246]	[3240] [3245]
K337		5,0000	2,1850	4,0000	1,7480	[2656]	[2658]
K338		10,0000	4,3611	9,0000	3,9250	[4167]	[4168]
K339		21,0000	11,1113	52,0000	27,5136	[2760]	[2263] [2721] [2745]
K340		85,0000	44,6650	189,0000	99,3139	[1777]	[1927] [1757] [2031]
K341		3,0000	1,4436	6,0000	2,8871	[625]	[626] [627]
K342		9,0000	3,8291	6,0000	2,5527	[2763]	[2765]
K343		7,0000	3,5161	9,0000	4,5208	[2265]	[2723] [2748]
K344		7,0000	3,4161	9,0000	4,3922	[2269]	[2728] [2753]
K345		6,0000	2,5393	5,0000	2,1161	[3875]	[3880]
K346		19,0000	8,0314	17,0000	7,1860	[601]	[602]
K347		4,0000	1,6889	4,0000	1,6889	[3450]	[3475]
K348		29,0000	13,7091	59,0000	27,8909	[2493]	[2467] [2474]
K349		7,0000	2,9277	6,0000	2,5095	[1808]	[1809]
K350		7,0000	2,9429	6,0000	2,5225	[2977]	[2931] [2955]
K351		27,0000	12,9464	51,0000	24,4543	[2988]	[2932] [2978]
K352		11,0000	5,4676	16,0000	7,9528	[2268]	[2725] [2750]
K353		15,0000	6,7223	22,0000	9,8594	[2181]	[3636] [1770] [2317] [2887]
K354		8,0000	3,2642	4,0000	1,6321	[3219]	[3691]
K355		6,0000	2,4300	4,0000	1,6200	[3500]	[3573]
K356		5,0000	2,0214	6,0000	2,4257	[4372]	[4376]
K357		7,0000	2,8200	7,0000	2,8200	[573]	[576]
K358		8,0000	3,2032	6,0000	2,4024	[877]	[893]
K359		9,0000	3,6000	15,0000	6,0000	[2128]	[2169] [2207] [1865] [2099]
K360		8,0000	3,2000	6,0000	2,4000	[3707]	[3822]
K361		6,0000	2,3914	4,0000	1,5943	[1995]	[2018]
K362		6,0000	2,3836	4,0000	1,5891	[3585]	[3588]
K363		6,0000	2,8630	6,0000	2,8630	[2266]	[2724] [2751]
K364		6,0000	2,7773	6,0000	2,7773	[2270]	[2726] [2749]
K365		6,0000	2,3800	6,0000	2,3800	[4462]	[4464]
K366		40,0000	15,7962	37,0000	14,6115	[3790]	[3791]
K367		6,0000	2,3571	6,0000	2,3571	[12]	[2428]
K368		11,0000	4,9694	18,0000	8,1318	[3101]	[3042] [3118] [3075] [3152]
K369		9,0000	3,5021	7,0000	2,7238	[3830]	[3832]
K370	Sim	73,0000	29,7945	67,0000	27,3457	[1699]	[1697] [1698]
K371		16,0000	6,4684	12,0000	4,8513	[373]	[724] [3215]
K372		6,0000	2,4216	6,0000	2,4216	[3524]	[3522] [3523]
K373		5,0000	1,9176	4,0000	1,5341	[1910]	[2009]
K374		7,0000	2,6756	5,0000	1,9111	[451]	[2427]
K375		6,0000	2,2886	4,0000	1,5257	[1797]	[2153]
K376		4,0000	1,5213	4,0000	1,5213	[3803]	[3804]
K377		9,0000	3,4204	9,0000	3,4204	[3259]	[3263]
K378		4,0000	1,6835	6,0000	2,5253	[4354]	[4363] [4378]
K379	Sim	97,0000	41,0703	94,0000	39,8001	[1384]	[1556] [1557] [1112] [1686]
K380		29,0000	13,4475	53,0000	24,5764	[3121]	[3048] [3104]
K381		5,0000	1,8872	4,0000	1,5098	[3558]	[3560]
K382		10,0000	4,0908	15,0000	6,1362	[2051]	[3687] [2520] [2227] [2250]
K383		5,0000	1,8795	5,0000	1,8795	[2469]	[2475]
K384		42,0000	17,5493	62,0000	25,9060	[1982]	[1980] [3114]
K385		22,0000	10,7882	18,0000	8,8267	[2201]	[2121] [2122] [2161] [2162]
K386		7,0000	3,8069	8,0000	4,3508	[2651]	[2628] [2633] [2715]
K387		14,0000	7,1129	25,0000	12,7015	[2770]	[2272] [2730] [2755]
K388		89,0000	43,5828	147,0000	71,9851	[1781]	[1922] [1756] [2026]
K389		5,0000	1,8487	4,0000	1,4790	[221]	[222]
K390		7,0000	2,5878	7,0000	2,5878	[218]	[228]
K391		4,0000	1,6795	6,0000	2,5192	[3813]	[3802] [3808]
K392		6,0000	2,2128	4,0000	1,4752	[3227]	[3256]
K393		8,0000	4,5841	8,0000	4,5841	[3900]	[2590] [2588] [2589]
K394		13,0000	4,7667	9,0000	3,3000	[819]	[820]
K395		8,0000	3,6965	20,0000	9,2413	[3729]	[3779] [3723] [3728]
K396		10,0000	4,6299	15,0000	6,9448	[2767]	[2267] [2727] [2752]
K397		7,0000	2,5270	4,0000	1,4440	[4433]	[4436]
K398		8,0000	2,8855	6,0000	2,1641	[706]	[719]
K399		15,0000	5,3437	12,0000	4,2750	[812]	[813]
K400		5,0000	1,7802	6,0000	2,1362	[3760]	[3764]
K401		23,0000	8,1129	34,0000	11,9930	[332]	[3221]
K402		8,0000	2,8160	7,0000	2,4640	[2256]	[2257]
K403		14,0000	4,9148	13,0000	4,5637	[1784]	[1930]
K404		12,0000	4,7709	11,0000	4,3734	[560]	[3396] [3866]
K405		16,0000	6,4838	10,0000	4,0524	[3789]	[3787] [3788]
K406		6,0000	2,0770	5,0000	1,7308	[738]	[739]
K407		31,0000	14,7342	70,0000	33,2708	[3540]	[3571] [740] [854]
K408		7,0000	2,4156	6,0000	2,0705	[3917]	[3919]
K409		174,0000	72,2363	139,0000	57,7060	[3907]	[3948] [3951] [3950] [3957]

Continua na próxima página ...

... continuação da página anterior						
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos
K410	Sim	207,0000	71,0567	170,0000	58,3558	[3938] [3939]
K411		5,0000	1,7135	4,0000	1,3708	[4186] [4219]
K412		7,0000	2,5548	8,0000	2,9198	[2916] [2526] [25] [36]
K413		9,0000	3,9399	9,0000	3,9399	[3770] [3732] [3733]
K414		23,0000	7,8400	19,0000	6,4765	[843] [2652]
K415		6,0000	2,0450	4,0000	1,3633	[2768] [2769]
K416		8,0000	2,7200	6,0000	2,0400	[4073] [4076]
K417		12,0000	5,1516	13,0000	5,5808	[623] [3860] [3862]
K418		9,0000	3,9120	8,0000	3,4773	[2616] [3482] [3483]
K419		11,0000	3,7138	7,0000	2,3633	[2000] [2001]
K420		8,0000	3,5601	7,0000	3,1151	[898] [896] [897]
K421		9,0000	3,5493	8,0000	3,1549	[575] [581] [590]
K422		8,0000	3,9222	15,0000	7,3542	[4162] [4137] [4133] [4616]
K423		4,0000	1,5298	9,0000	3,4421	[4382] [4391] [4380] [4381]
K424		10,0000	3,9737	10,0000	3,9737	[2028] [1755] [1925]
K425		9,0000	3,0191	7,0000	2,3482	[891] [892]
K426		5,0000	1,6752	4,0000	1,3401	[3544] [3548]
K427		12,0000	4,0057	8,0000	2,6705	[3535] [3538]
K428		8,0000	2,6514	6,0000	1,9886	[1812] [2620]
K429		5,0000	1,6538	5,0000	1,6538	[3456] [3470]
K430		77,0000	29,5044	74,0000	28,3549	[1458] [1330] [1640]
K431		10,0000	3,3048	8,0000	2,6438	[1744] [1746]
K432		9,0000	3,1668	7,0000	2,4631	[4023] [4021] [4022]
K433		9,0000	2,9640	7,0000	2,3053	[3369] [3394]
K434		10,0000	3,2833	5,0000	1,6417	[356] [357]
K435		4,0000	1,3120	4,0000	1,3120	[4061] [4065]
K436		15,0000	6,1176	16,0000	6,5255	[3078] [3047] [3107]
K437		6,0000	1,9654	4,0000	1,3103	[404] [409]
K438		11,0000	3,5674	11,0000	3,5674	[3459] [3462]
K439		6,0000	1,9457	4,0000	1,2971	[3233] [3234]
K440		6,0000	2,2982	6,0000	2,2982	[3765] [3766] [3769]
K441		6,0000	1,9442	4,0000	1,2962	[3371] [3373]
K442		5,0000	1,8355	6,0000	2,2026	[364] [176] [177]
K443		34,0000	10,9178	32,0000	10,2756	[3460] [3461]
K444		7,0000	2,7731	8,0000	3,1692	[3271] [3287] [3297] [3304]
K445		6,0000	1,9226	5,0000	1,6022	[4425] [4430]
K446		5,0000	1,8162	7,0000	2,5427	[2676] [2677] [2678]
K447		13,0000	4,9350	10,0000	3,7962	[1895] [1864] [2168]
K448		5,0000	1,5951	4,0000	1,2760	[3464] [3474]
K449		15,0000	4,7824	16,0000	5,1012	[2452] [2505]
K450		8,0000	2,9313	7,0000	2,5649	[2057] [2059] [2060]
K451		13,0000	5,1186	9,0000	3,5436	[2973] [2947] [2974]
K452		44,0000	19,4563	59,0000	26,0892	[3913] [3914] [3906] [3909] [3912] [3904] [3910]
K453		9,0000	2,8309	8,0000	2,5164	[413] [439]
K454		5,0000	1,6240	8,0000	2,5984	[3797] [3784] [3809]
K455		92,0000	43,1541	446,0000	209,2038	[1873] [2107] [2136] [2215] [2097] [1863] [2167] [2126] [2205] [1780] [1906] [2231]
K456		8,0000	4,3851	16,0000	8,7703	[88] [77] [79] [80] [81] [76] [78]
K457		7,0000	2,3622	7,0000	2,3622	[4154] [4155] [4159]
K458		8,0000	2,5013	6,0000	1,8760	[4016] [4017]
K459		13,0000	4,9724	17,0000	6,5024	[4371] [4351] [4366] [4370]
K460		12,0000	3,7486	8,0000	2,4991	[3714] [3715]
K461		9,0000	2,7933	7,0000	2,1725	[2355] [2356]
K462		30,0000	12,6226	100,0000	42,0755	[1298] [1300] [1297] [1302] [1299] [1301]
K463		16,0000	6,8386	24,0000	10,2579	[2957] [2959] [2929] [2980]
K464		30,0000	10,4286	21,0000	7,3000	[3610] [3608] [3609]
K465		8,0000	2,4582	6,0000	1,8436	[1981] [1985]
K466		34,0000	15,4955	33,0000	15,0398	[3095] [1966] [3072] [3073] [3076]
K467		14,0000	5,2676	10,0000	3,7626	[375] [3239] [3244]
K468		9,0000	3,1401	6,0000	2,0934	[707] [720] [4046]
K469		8,0000	2,8615	8,0000	2,8615	[2759] [2262] [2744]
K470		63,0000	19,1673	61,0000	18,5588	[919] [1385]
K471		11,0000	3,3395	6,0000	1,8215	[4037] [4044]
K472		7,0000	2,1223	5,0000	1,5159	[3870] [3872]
K473		5,0000	2,0786	7,0000	2,9100	[4608] [4138] [4611]
K474		16,0000	4,8421	9,0000	2,7237	[2857] [2860]
K475	Sim	11,0000	3,3266	7,0000	2,1169	[3698] [3816]
K476		7,0000	3,0017	7,0000	3,0017	[3270] [3268] [3269]
K477		12,0000	4,8654	26,0000	10,5417	[3312] [3364] [3325] [3353]
K478		8,0000	2,9537	7,0000	2,5845	[3974] [3970] [3973]
K479		12,0000	3,6040	10,0000	3,0033	[1947] [2954]
K480		8,0000	3,3094	12,0000	4,9641	[4203] [4177] [4175] [4214]
K481		18,0000	5,3940	11,0000	3,2963	[87] [2290]
K482		20,0000	7,0844	21,0000	7,4387	[65] [62] [63]
K483		4,0000	1,1824	4,0000	1,1824	[2627] [2638]
K484		8,0000	3,7343	18,0000	8,4023	[292] [330] [342] [343] [308] [309]
K485		7,0000	2,0667	5,0000	1,4762	[3513] [3514]
K486		21,0000	8,1516	41,0000	15,9151	[3065] [3110] [3043] [3102] [3119] [3156]
K487		49,0000	23,4497	147,0000	70,3490	[325] [333] [326] [327]
K488		7,0000	2,0550	4,0000	1,1743	[2515] [2516]
K489		7,0000	2,0417	7,0000	2,0417	[4210] [4267]
K490		15,0000	4,3647	10,0000	2,9098	[3692] [3693]
K491		8,0000	2,3200	6,0000	1,7400	[884] [885]
K492		6,0000	1,7359	7,0000	2,0252	[2461] [2496]
K493		19,0000	7,3927	15,0000	5,8363	[883] [888] [886] [889]
K494		5,0000	1,4293	4,0000	1,1435	[3767] [3768]
K495		10,0000	3,8132	10,0000	3,8132	[677] [696] [697]
K496		7,0000	2,3452	6,0000	2,0102	[2560] [2545] [2561]
K497		8,0000	2,8260	8,0000	2,8260	[3780] [3782] [3798]
K498		6,0000	1,7010	7,0000	1,9845	[705] [718]
K499		11,0000	3,1843	8,0000	2,3159	[598] [582] [621]

Continua na próxima página ...

... continuação da página anterior							
Grupo	IT?	GSRank	GSIRank	GFRank	GFIRank	Métodos	
K500		6,0000	1,6980	4,0000	1,1320	[2848]	[2853]
K501		15,0000	4,2424	12,0000	3,3939	[2695]	[2700]
K502		11,0000	3,9102	11,0000	3,9102	[3231]	[3264] [3252] [3253]
K503		7,0000	1,9695	5,0000	1,4068	[3485]	[3486]
K504		6,0000	1,6975	6,0000	1,6975	[4374]	[4353] [4390]
K505		13,0000	3,6400	10,0000	2,8000	[1953]	[1954]
K506		66,0000	18,4751	50,0000	13,9963	[3958]	[3959]
K507		6,0000	1,6788	4,0000	1,1192	[2615]	[2624]
K508		4,0000	1,1181	4,0000	1,1181	[2348]	[2357]
K509		6,0000	1,6753	4,0000	1,1169	[3648]	[3649]
K510		5,0000	2,5144	10,0000	5,0288	[1417]	[1467] [1469] [1470]
K511		6,0000	2,5140	6,0000	2,5140	[2397]	[2394] [2396]
K512		9,0000	3,2283	8,0000	2,8696	[1806]	[1796] [1801] [1803]
K513		14,0000	5,0163	24,0000	8,5993	[3623]	[3611] [3654]
K514		19,0000	7,7682	27,0000	11,0390	[3128]	[3171] [3046] [3106]
K515		6,0000	1,6445	4,0000	1,0964	[580]	[606]



## APÊNDICE D

### PONTOS DE CORTE IDENTIFICADOS

Este apêndice apresenta a lista de pontos de corte identificados pelo método ARPIM respectivamente nos sistemas JHotDraw, Tomcat e HSQLDB. Para cada grupo identificado pela instância 12 da fase de agrupamento foram listados os pontos de corte identificados e os respectivos aspectos onde foram refatorados.

Na Tabela D.1 são apresentados os pontos de corte identificados pelo ARPIM no sistema JHotDraw. Na Coluna Grupo são listados todos os grupos identificados pela instância 12 da fase de agrupamento que possuem métodos pertencentes as instâncias de interesses transversais validadas, sendo que os métodos pertencentes ao grupo podem ser consultados na Tabela C.2. Na Coluna Pontos de Corte são listados os pontos de corte identificados para o respectivo grupo, sendo que são apresentados somente os identificadores dos métodos. Os identificadores podem ser consultados na Tabela C.1. Os grupos que não possuem pontos de corte identificados possuem a Coluna Pontos de Corte igual a vazio ( $\emptyset$ ). A Coluna Aproveitado? indica o aspecto onde o ponto de corte foi refatorado. Caso o ponto de corte não tenha sido refatorado, a Coluna Aproveitado é igual a Não. Caso o grupo não possua pontos de corte, a Coluna Aproveitado é igual a N/A.

Tabela D.1: Pontos de Corte identificados pelo ARPIM no sistema JHotDraw

Grupo	Ponto de Corte	Aproveitado?
K136	3064 ANTES DA EXECUCAO [1066,1135,2509,715,883,2037,2071,2535,2467,2292,2109,1300,2013,1148,1442,2245,1179,3061,1317]	UndoRedoActivity.aj
K137	1545 ANTES DA CHAMADA 1067	PasteCommandUndo.aj
K147	3060 ANTES DA EXECUCAO [851,1134,2307,2508,714,882,1065,2036,1428,2070,1299,2534,2466,2291,2012,2108,1441,2218,1147,2244,1178,1316]	UndoableCommand.aj
K166	1551 APOS A EXECUCAO [2202,2066,2007,2504,2315,2530,1437,2287,1173,2461,1143,2058,724,2240]	CommandContracts.aj
K168	$\emptyset$	N/A
K170	$\emptyset$	N/A
K171	1571 APOS A EXECUCAO [2302,442,2162,2262,2313,317,2012]	Não
K175	1910 ANTES DA EXECUCAO [1913,1912,1905,1792]	SelectionChangedNotification.aj
K190	2842 APOS A EXECUCAO [2865,3076]	Não
K232	3061 ANTES DA EXECUCAO [2219,2308,1429,852]	Não
K256	1486 APOS A EXECUCAO [2374,2177,2574]	Não
K259	1912 ANTES DA CHAMADA 806	Não
	1912 ANTES DA CHAMADA 1362	Não
	1912 ANTES DA EXECUCAO [805,1268,684,683,1890,486,1892,1199,1198,862,1361]	Não
	1371 APOS A EXECUCAO [1361,1394]	Não
	1913 APOS A EXECUCAO [684,822,683,1262,1199,862,1198,805,1268,1267,1835,1890,486,1892]	Não
K331	1925 ANTES DA EXECUCAO [2279,1286,2169,1047]	PersistentFigure.aj
	1047 ANTES DA EXECUCAO [1169,1116,1385,1343,1329,815,690]	PersistentAttributeFigure.aj
K332	1926 ANTES DA EXECUCAO [1048,2280,2170,1287]	PersistentFigure.aj
	1048 ANTES DA EXECUCAO [1117,1170,816,1386,1330,691,1344]	PersistentAttributeFigure.aj
K334	$\emptyset$	N/A
K354	1860 ANTES DA EXECUCAO [2674,2066,2007,2504,2315,2976,2530,1437,2287,2223,1173,2461,1143,962,2058,3143,2240]	CommandContracts.aj
K361	$\emptyset$	N/A

Na Tabela D.2 são apresentados os pontos de corte identificados pelo ARPIM no sistema Tomcat. Na Coluna Grupo são listados todos os grupos identificados pela instância 12 da fase de agrupamento que possuem métodos pertencentes as instâncias de interesses transversais validadas, sendo que os métodos pertencentes ao grupo podem ser consultados na Tabela C.4. Na Coluna Pontos de Corte são listados os pontos de corte identificados para o respectivo grupo, sendo que são apresentados somente os identificadores dos métodos. Os identificadores podem ser consultados na Tabela C.3. Os grupos que não possuem pontos de corte identificados possuem a Coluna Pontos de Corte igual a vazio ( $\emptyset$ ). A Coluna Aproveitado? é igual a N/A em todos os casos devido ao fato que o Tomcat não possui versão orientada a aspectos.

Tabela D.2: Pontos de Corte identificados pelo ARPIM no sistema Tomcat

Grupo	Ponto de Corte	Aproveitado?
K52	2157 ANTES DA EXECUCAO [2184,2174,2159,2173,2183]	N/A
K66	2644 APOS A EXECUCAO [2641,2975,2625,3337,3341,2992,2636,2618,3047]	N/A
K103	5046 ANTES DA EXECUCAO [5379,5408,5377]	N/A
K140	∅	N/A
K177	7408 ANTES DA CHAMADA 1541 7408 APOS A CHAMADA 5299	N/A N/A
K181	∅	N/A
K197	∅	N/A
K203	7993 ANTES DA EXECUCAO [7982,7976,7979,7975,7984] 7993 APOS A EXECUCAO [7982,7976,7979,7975,7984]	N/A N/A
K223	∅	N/A
K237	∅	N/A
K239	∅	N/A
K245	∅	N/A
K273	∅	N/A
K277	∅	N/A
K291	∅	N/A
K292	∅	N/A
K294	∅	N/A
K300	∅	N/A
K302	2622 ANTES DA CHAMADA 2625	N/A
K305	2621 ANTES DA CHAMADA 7806	N/A
K309	∅	N/A
K312	7578 APOS A EXECUCAO [7755,4082,3160,313,384,981]	N/A
K316	∅	N/A
K318	10 ANTES DA CHAMADA 11 10 ANTES DA EXECUCAO [274,272,276,280,47,284,288,258,292,55,250,83,260,246,264,268] 11 APOS A EXECUCAO [258,274,272,292,250,55,260,276,246,268] 11 APOS A CHAMADA 10	N/A N/A N/A N/A
K329	2617 APOS A EXECUCAO [3144,3189]	N/A
K355	∅	N/A
K393	∅	N/A
K506	7868 APOS A EXECUCAO [7870,7868]	N/A
K514	∅	N/A
K518	∅	N/A
K522	2632 ANTES DA EXECUCAO [3150,2635] 4967 ANTES DA EXECUCAO [4825,4917,4857,5002,4933,4786] 3065 APOS A EXECUCAO [3704,3062]	N/A N/A N/A
K525	∅	N/A
K530	5413 ANTES DA EXECUCAO [4206,1121] 4966 ANTES DA EXECUCAO [4916,4824,4856,5001,4932,4785] 2631 APOS A EXECUCAO [3149,3195]	N/A N/A N/A
K542	∅	N/A
K564	∅	N/A
K575	8854 APOS A EXECUCAO [8856,8855]	N/A
K629	7407 ANTES DA EXECUCAO [7710,823,3912,7648,5862,4464,4297,3282,3253,378,1745,5504,322,4964,2629,7684,7608, 992,5411,5383,4093,1515,3215,7750] 7406 ANTES DA EXECUCAO [7709,5503,822,4296,991,3281,3252,5861,377,3911,1744,4963,2628,7607,321,7647,4463, 5410,7749,4092,5382,7683,1514,3214] 7407 APOS A EXECUCAO [7710,823,3912,7648,5862,4464,4297,3282,3253,378,1745,5504,322,4964,2629,7684,7608, 992,5411,5383,4093,1515,3215,7750] 7406 APOS A EXECUCAO [7709,5503,822,4296,991,3281,3252,5861,377,3911,1744,4963,2628,7607,321,7647,4463, 5410,7749,4092,5382,7683,1514,3214]	N/A N/A N/A N/A

Na Tabela D.3 são apresentados os pontos de corte identificados pelo ARPIM no sistema HSQLDB. Na Coluna Grupo são listados todos os grupos identificados pela instância 12 da fase de agrupamento que possuem métodos pertencentes as instâncias de interesses transversais validadas, sendo que os métodos pertencentes ao grupo podem ser consultados na Tabela C.6. Na Coluna Pontos de Corte são listados os pontos de corte identificados para o respectivo grupo, sendo que são apresentados somente os identificadores dos métodos. Os identificadores podem ser consultados na Tabela C.5. Os grupos que não possuem pontos de corte identificados possuem a Coluna Pontos de Corte igual a vazio ( $\emptyset$ ). A Coluna Aproveitado? indica o aspecto onde o ponto de corte foi refatorado. Caso o ponto de corte não tenha sido refatorado, a Coluna Aproveitado é igual a Não. Caso o grupo não possua pontos de corte, a Coluna Aproveitado é igual a N/A.

Tabela D.3: Pontos de Corte identificados pelo ARPIM no sistema HSQLDB

Grupo	Ponto de Corte	Aproveitado?
K18	∅	N/A
K50	∅	N/A
K51	∅	N/A
K204	∅	N/A
K213	∅	N/A
K225	∅	N/A
K231	4069 ANTES DA EXECUCAO [4175,4091,4219] 4069 APOS A EXECUCAO [4177,4175,4219]	DatabaseManagerSwing.aj DatabaseManagerSwing.aj
K238	4198 APOS A EXECUCAO [4214,4203]	DatabaseManagerSwing.aj
K311	1696 APOS A EXECUCAO [1687,1699,1698,1697,1393]	jdbcStatement.aj

Continua na próxima página ...

... continuação da página anterior		
Grupo	Ponto de Corte	Aproveitado?
K314	∅	N/A
K319	∅	N/A
K370	1698 APOS A CHAMADA 1113	Não
	1697 ANTES DA EXECUCAO [1686,1384,1112,1557,1556]	jdbcConnection.aj
	1697 APOS A EXECUCAO [1686,1384,1112,1557,1556,1467,1469]	jdbcConnection.aj
K379	1698 APOS A EXECUCAO [1117,1116,1640,1458,1385,1115,919,1330]	Não
	1557 ANTES DA EXECUCAO [1467,1469,1470]	jdbcResultSet.aj
K410	1686 ANTES DA EXECUCAO [1385,919]	jdbcPreparedStatement.aj
	3938 APOS A EXECUCAO [3316,2543,168,2782,3906,3939,2784,602,2544,3365,2584]	Não
K475	3939 DURANTE A EXECUCAO 718	Não
	∅	N/A

EDISON KLAFKE FILLUS

**CAAMPI: UMA ABORDAGEM BASEADA EM TÉCNICAS  
DE AGRUPAMENTO PARA MINERAÇÃO DE ASPECTOS  
E IDENTIFICAÇÃO DE PONTOS DE CORTE**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientadora: Profa. Dra. Silvia Regina Vergilio

CURITIBA

2012